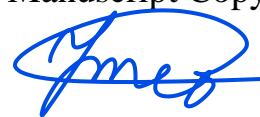


Satbayev University

UDC 534:004.932.72'1(043)

Manuscript Copyright



UTEBAYEVA DANA ZHOLDYBAYKYZY

Research of effective UAV detection using acoustic data recognition

6D071900 – Radio engineering, Electronics and Telecommunications

Thesis for the Degree of
Doctor of Philosophy (PhD)

Supervisors
candidate of technical sciences,
associate professor
L.B. Ilipbayeva

doctor PhD,
professor
E.T. Matson
(Purdue University)

Republic of Kazakhstan
Almaty, 2023

CONTENTS

NORMATIVE REFERENCES	4
SYMBOLS AND ABBREVIATIONS	5
INTRODUCTION	6
1 STATE OF THE ART: UAV DETECTION WITH ACOUSTIC DATA	10
1.1 UAV detection systems.....	10
1.2 Acoustic data-based UAV detection.....	12
1.2.1 The Role of Classification for UAV Acoustic Data Recognition.....	14
1.3 Related works on UAV sound detection and classification methods...	15
1.3.1 Pre-processing methods of the UAV acoustic data recognition system	18
1.3.2 Machine Learning algorithms for UAV acoustic data recognition.....	20
1.3.3 Deep learning algorithms for acoustic data recognition.....	21
1.4 Problem Statement: The protection system for strategic areas from unidentified UAVs based on acoustic recognition.....	27
1.4.1 Suspicious UAVs with high-risk cases: Loaded and Unloaded UAVs	27
1.4.2 UAV distance Identification.....	28
1.4.3 Multiple model UAV recognition.....	28
2 UAV ACOUSTIC DATA PREPARATION	29
2.1 UAV sound recording in different positions and models.....	29
3 MATHEMATICAL VIEW ON THE SIGNAL PRE-ANALYSIS STEP IN TIME AND FREQUENCY DOMAINS	32
3.1 Foundational principle of sound data representation.....	32
3.2 Acoustic Data in Time Domain.....	34
3.3 Short-Time Fourier-Transform (STFT).....	36
3.4 Mel-Scale Spectrograms.....	39
3.5 An efficient signal processing proposal: the KAPRE method.....	40
4 DEEP LEARNING METHODS FOR UAV ACOUSTIC DATA RECOGNITION	42
4.1 Convolutional Neural Networks (CNNs) in Sound Recognition Problems.....	43
4.2 Recurrent Neural Networks (RNNs) in Sound Recognition.....	45
4.2.1 Simple Recurrent Neural Networks (RNNs) in Sound Recognition...	45
4.2.2 Long-term short-term memory (LSTM) for sound recognition.....	46
4.2.2.1 Bidirectional Long Short-Term Memory (LSTM).....	49
4.2.3 Gated Recurrent Neural Networks (GRU) for Sound Recognition.....	50
5 REAL-TIME UAV ACOUSTIC DATA RECOGNITION AND CLASSIFICATION SYSTEM	52
5.1 The proposed real-time Drone Sound recognition system.....	52
5.1.1 Adaptation of UAV Sound Recordings for Real Time System.....	52
5.1.2 Processing of UAV acoustic signals using the KAPRE method: Melspectrogram.....	56
5.1.3 Real-time and RNN network-based UAV sound recognition architecture.....	58
5.2 Results and discussion of the Proposed System.....	62

CONCLUSION	71
REFERENCES	72
APPENDIX A – Model and layers of the CNN algorithm based on the publication.....	78
APPENDIX B – Visualization of the Stacked BiLSTM-CNN model presented in publication.....	79
APPENDIX C – Composition of the initial dataset of the study.....	80
APPENDIX D – Studying the sounds of background objects and UAVs with 6 classes in the Time domain.....	81
APPENDIX E – Experimental studies at the stage of audio data adaptation...	82
APPENDIX F – Plot of the Power level of UAV sound signals.....	83
APPENDIX G – Investigation of spectrograms with various hyperparameters during the experiment	84
APPENDIX H – Implementation of the proposed system in the Python program.....	85
APPENDIX I – Confusion matrix from an experiment on recognizing UAVs at close range and a certain state.....	86
APPENDIX K – Conducting experimental studies at international research institutions.....	87
APPENDIX L – Publication of experimental studies at the conference.....	90
APPENDIX M – Minute on the acceptance of a scientific project by the "Zhas Galym 2022-2024"	91

NORMATIVE REFERENCES

This thesis uses references to the following standards:

“Instructions for the preparation of a dissertation and author’s abstract”
Ministry of education and science of the Republic of Kazakhstan, 28th September,
2004. 377-3 y.

GOST 7.32-2001. Report on research work. Structure and design rules.

GOST 7.1-2003. Bibliographic record. Bibliographic description. General
requirements and compilation rules.

GOST 7.32-2017. System of standards of information, librarianship and
publishing. Research report. Structure and design rules.

SYMBOLS AND ABBREVIATIONS

UAV	– Unmanned aerial vehicles
UAS	– Unmanned aerial systems
R&D	– research and development
CNN	– Convolutional Neural Network
RNN	– Recurrent Neural Network
LSTM	– Long-Short Term Memory
BiLSTM	– Bidirectional Long-Short Term Memory
GRU	– Gated Recurrent Unit
DL	– Deep Learning
ML	– Machine Learning
FT	– Fourier Transform
DFT	– Discrete Fourier Transform
FS	– Fourier Series
STFT	– Short-time Fourier transform
MFC	– Mel Frequency Cepstral
MFCC	– Mel Frequency Cepstrum Coefficients
WAV	– Waveform Audio File
AAC	– Advanced Audio Coding
FLAC	– Free Lossless Audio Codec
MSS	– Mel-Scale Spectrograms
KAPRE	– Keras Audio Preprocessing Layers
CPU	– central processing unit
GPU	– Graphics Processing Unit
NLP	– natural language processing
Fmaps	– feature maps

INTRODUCTION

General characteristics of the work. The proposed dissertation researches the development of a UAV detection system based on the recognition of acoustic signatures. The recognition of UAV acoustic signals was processed in the form of Melspectrogram frequency characteristics and studied by deep learning methods, in particular, recurrent neural networks. As a result, the GRU based architecture in the developed structure was proposed as an effective method for the UAV acoustic data recognition system.

Relevance of the work. In recent years, unmanned aerial vehicles have become widespread and have become very popular. The use of these small devices, also called drones, is increasing day by day, especially for purposes such as children's toys, adult recreational entertainment, photography, video surveillance in hard-to-reach places, agronomy, military intelligence, delivery, and transportation. Its increased technological capacity, including longer flight times, the ability to take flexible photos and videos from a variety of angles, and the opportunity for unfettered entry into different zones, is the primary factor behind their increased use by people. [1-7]. Also, over the past decade, the mass manufacturing of unmanned aerial vehicles (UAVs) at affordable prices has led to the problem of continuous use of these vehicles for various dubious and recreational purposes [1, p. 862-864; 2, p. 242-243; 6, p. 138669-138680; 7, p. 3856-1-3856-17]. The use of these vehicles carelessly or destructively puts individuals, their lives, protected institutions, and international borders in danger. These justifications take into account the fact that UAVs are becoming increasingly hazardous. Recently, there have been several drone incidents in the country. In particular, drones of the Republic of Uzbekistan, one of the five countries bordering our country, were registered in the border areas during their unauthorized flights. At the same time, the headquarters of the Border Guard National Security Troops of the Republic of Kazakhstan confirmed the fact of the UAV crash of the Republic of Uzbekistan, and negotiations were held between the relevant services of the Republic of Kazakhstan and the Republic of Uzbekistan [8]. Another similar incident was observed in 2019 in Nur-Sultan, above the building of the Ministry of Defense of the Republic of Kazakhstan. This is not the first case of an underwater incident with the unauthorized use of a UAV. At the same time, two people who tried to launch the quadcopter were detained and prosecuted. As a result, the procedure for using unmanned aerial vehicles (quadcopters) over settlements is published on the official website of the Ministry of Industry and Infrastructure Development of the Republic of Kazakhstan [9, 10]. In addition, the UAV was found in the south of the country in the city of Arys [11]. Moreover, drones used for recreational purposes and as children's toys have caused significant damage around the world. That is, there are many cases of damage caused by improper management. At the Saudi Arabian border city of Asir, a similar unmanned aerial vehicle (UAV) disaster also happened [12]. Another UAV disaster took place in China: 12 of the 200 drones that were in the air at the same time as a light show in Zhengzhou crashed. Only 2.5 minutes after their triumphant climb, the UAVs in this incident

started to plummet back to the ground, colliding with everything in their path as they dropped, including trees, automobiles, and other objects. Unmanned drones forced several event attendees to escape and hide [13]. Overall, the reasons why the sustainable use of these vehicles is impossible are more thoroughly addressed in works [6, p. 138670-138681; 7, p. 3856-1-3856-17]. Thus, the high frequency of unauthorized drone flights requires the development of reliable real-time drone detection systems for protected areas. Therefore, these detection systems for the unauthorized use of UAVs are becoming more and more *relevant*. Particularly in the buildings of establishments such as kindergartens, hospitals, universities, administrations, ministries, border regions of the nation, protected territories where military bases are situated, reservoirs that shield major cities from snowmelt, and agricultural areas. In order to stop the proliferation of unlicensed drones in restricted and protected key regions, there has been an increasing demand for research into security measures based on drone detection systems. Generally, in the UAV research and development (R&D) market, the Drone Detection System is being studied based on the following four main methods: Radar reconnaissance, Camera-based detection, RF signal processing, and Acoustic Sensor listening Detection [6, p. 138675; 7, p. 3856-2]. The aforementioned drone incidents require the preparation of a recognition system, including the recognition of their position, load states and models during flight according to their protection zones. That is why the acoustic direction is *relevant* due to its technical capabilities for recognizing such extended tasks.

The goal of the research. The goal of this thesis is to investigate an efficient recognition method of UAV Acoustic Data.

The Objectives of the research. In the studies of this dissertation, three main objectives are set:

1. Preparation and adaptation of UAV acoustic data and their various states.
2. Develop an efficient real-time system that integrates the acoustic signal processing step into the deep learning architecture.
3. Explore UAV acoustic data recognition using deep learning networks such as CNN, SimpleRNN, LSTM, BiLSTM and GRU.

In the first objective, UAV sounds were recorded in different states such as "Unloaded" and "Loaded". The sounds of the loaded UAVs were recorded in the state when they had a payload during the flight with different weights. The sounds of the environment in various scenarios and objects with increased motor noise were recorded as "background noise". Various series of UAV models were also recorded.

At the second objective, the processing of UAV acoustic signals was combined into the architecture of deep learning networks.

In the third objective, deep learning algorithms were studied. In particular, all types of recurrent neural networks such as SimpleRNN, LSTM, BiLSTM and GRU. These neural networks have been investigated and applied to UAV sound detection. A comparative analysis of CNN and GRU networks was also carried out, as a result, the advantage of the GRU network for recognizing UAV acoustic data was determined.

Methods of the research. The research of this thesis was carried out on the basis of a combination of analytical and empirical methods. In particular, the experimental approach was employed to collect UAV sounds for the study's first objective. Additionally, Fast Fourier analysis, Short-Time Fourier Transform and Mel spectrogram filters were employed to analyze the audio signals that were gathered. Moreover, the Convolutional Neural Networks (CNNs) and Recurrent Neural networks (RNNs) deep learning methods were extensively used to achieve the last objective.

The scientific novelty of the work.

The novelty of this study is to development of an architecture of a UAV acoustic data recognition system with the integration of a modified Melspectrogram.

The theoretical and practical significance of the work. In this dissertation work, types of recurrent neural networks for recognition of UAV acoustic data were extensively investigated. The proposed system is recommended for national security systems, in particular the security of people, densely populated areas, airports, government buildings, kindergartens, schools, universities, national borders, customs and strategic places.

Research publications:

1. Multi-label UAV sound classification using Stacked Bidirectional LSTM // 2020 Fourth IEEE International Conference on Robotic Computing (IRC), (Taichung, 2020. – P. 453-458).

2. Stacked BiLSTM - CNN for Multiple label UAV sound classification // 2020 Fourth IEEE International Conference on Robotic Computing (IRC), (Taichung, 2020. – P. 470-474).

3. Effectiveness of the System of Unmanned Aerial Vehicles Detection on the Basis of Acoustic Signature // Vestnik KazNRTU. Vestnik KazNRTU. – 2020. – Vol. 4, Issue 140. – P. 300-307 (ISSN1680-9211).

4. Investigation of Acoustic Signals in Uav Detection Tasks for Various Models (2021-08-17).

5. Survey on Different Drone Detection methods in the Restricted Flight Areas // Vestnik KazNRTU. – 2019 (ISSN1680-9211).

6. Practical Study of Recurrent Neural Networks for Efficient Real-Time Drone Sound Detection: A Review // Drones. – 2023. – №7. – P. 26.

Acknowledgments. I would like to thank all who supported me with this research and the writing of this research thesis, especially Professors L. Iipbayeva and E. Matson, who guided me toward a systematic approach to experimentation, Professor John S. Gallagher, who originally provided the UAV audio data, and my lab-mate U. Seidaliyeva, who worked on a related project in Vision-based UAV detection and gave moral support during the study. I would like to express my deep gratitude to my family and the memory of my mother Gulzhamila Utebayeva, who has always inspired me and provided the basis for ambitious goals in science.

Structure and scope of the thesis.

This dissertation consists of 5 parts: "State of the Art: UAV detection with acoustic data", " UAV acoustic data preparation", "Mathematical view on the signal

pre-analysis step in time and frequency domains", "Deep Learning methods for UAV acoustic data recognition" and "Real-time UAV acoustic data recognition and classification system".

1 STATE OF THE ART: UAV DETECTION WITH ACOUSTIC DATA

1.1 UAV detection systems

Currently, UAVs, also referred to as drones, are becoming more and more popular among consumers as they provide an easy solution for most day-to-day needs. They are being used increasingly in areas such as agriculture, photography, film production, law enforcement, logistics, and transportation. Drones are very useful because they can reach even the most remote places without pilot control on the board. With this capability, drones have become a global phenomenon and they are increasingly being used for many activities. Modern technology is evolving more quickly than it has ever been. Our lifestyles are constantly changing due to advancements that could improve matters, quicker, and simpler. In this regard, UAVs have completely changed the aviation industry, making it safer, more accessible, and much more productive. Drones are getting more compact and accessible as technology develops. One UAV can actually now fit in a human hand. They are currently being used for various reasons by various companies due to their modest size. Some types of them are used to support search and rescue efforts. Drones are growing in popularity in the commercial sector as well since they can be simpler to use, safer to operate, and more reasonably priced. They are increasingly being used for military purposes due to their long stay in the air. They are also becoming more and more popular with clients and professionals looking for advanced alternatives for their homes and business. Although these UAVs were originally created and developed for military use, today they serve an important purpose for various businesses, governments, and individuals around the world [14]. Despite receiving a lot of attention in a variety of civil and commercial solutions, UAVs pose a sort of airspace security threat that can put in danger individuals, property, key areas, and buildings. While the targets and complexity of such threats can range from incompetent skill of the UAV controller to intentional attacks, they are all capable of causing significant damage [15, 16]. There have been many cases of UAV penetration into such protected areas and objects. Particularly, a drone crashed in front of the White House lawn in the United States in 2015 due to a drunken government official, in Canada in 2017 there was a small plane crash with a drone, and in London in 2018, thousands of passengers were forced to cancel flights due to a suspicious drone flying over Gatwick Airport [7, p. 3856-1-3856-2]. Several airports in the US, UK, Ireland and the UAE experienced major outages in the first few months of 2019 as a result of drone detections [17]. Drones are also often used for delivery systems, terrorist activities, transportation of criminals, unauthorized use of goods in restricted areas such as customs and prisons, among many other things [1, p. 862-863; 2, p. 242-243; 18, 19]. Drones can also be used to take photos or videos from unusual angles using additional cameras. All these reasons have made it relevant to recognize UAVs when they have additional load [20]. This shows that the problem of not only detecting a drone, but also simultaneously determining its state in relation to loads is relevant. In an attempt to protect people from all these risks, it is essential to develop a preventive strategy. For the booming UAV business to avoid

abuse and unauthorized use, appropriate regulations and rules must evolve at the same pace as technology [1, p. 862-865; 2, p. 243-244; 4, p. 2-3]. Air traffic controllers around the world are working hard to reduce the chance of unauthorized drone use. These rules and regulations may discourage careless or unskilled drone piloting, but cannot prevent criminal or terrorist attacks in protected areas. To form an effective approach to this problem, it is necessary to develop technologies that provide: 1) detection, classification, and tracking of drones, 2) suppression of drones, and 3) collection of evidence on the fact of violation [6, p. 138669]. The scope of work on all three tasks is extensive and, in turn, requires individual research approaches [6, p. 138670]. And among these objectives, the detection of drones is a significant research issue and direction. The detection of these objects is the first issue even for the development of preventive measures against the use of drones for inept and malicious purposes. Therefore, it is vital to create a system of UAV detection. Another significant reason why the issue of detecting the presence of UAVs in protected areas is becoming relevant today is the use of UAVs for military-political purposes at the borders of countries [21, 22]. In general, scientists and technical solutions have developed four main drone detection methods such as radar, acoustic sensor, visual and radio frequency (RF) signals-based detection [6, p. 138678]. In the last decade, general object detection has been widely used by artificial intelligence methods, in particular Machine learning and Deep learning methods. This is due to the fact that these methods can achieve high results in the accuracy of object recognition. Also, the main four methods for detecting unmanned aerial objects began to be studied based on these machine learning and deep learning methods. And now, if we focus more closely on the task of detection, drones are available in different positions, states and models, depending on the purpose of their use. If the problem of recognition creates a high level of demand, the question of *UAV classification* to determine their status and types is also put forward. And these two tasks are often carried out simultaneously in terms of technical solutions. In the task of detecting and classifying drones, Radar technology is considered as a sensor that provides accurate identification of a flying object at a long distance and performance independent of environmental factors and light levels. Small commercial UAVs flying at relatively low speeds along non-ballistic trajectories cannot be detected by the radar, since it is primarily designed to identify high-speed targets with ballistic trajectories, such as military drones and missiles [7, p. 138678-138679; 23]. Radar sensors are often used as reliable means of detection, but their classification capabilities are unsatisfactory [24]. When a classification problem arises, the similarities between the key characteristics of UAVs and birds often make it difficult to identify their differences, making this option ineffective. In addition, the price is quite exorbitant. In addition, researchers have begun to show great interest in the *acoustic sensor* approach to drone detection. This method is considered to be a cost-effective detection system that, using arrays of acoustic sensors or microphones, can recognize the distinctive sound characteristics of UAV rotors even in poor visibility condition. Machine learning and Deep learning-based acoustic identification of drones are new advances in drone detection research. A significant barrier to this

study is the lack of enough data on multiple drone models flying at different altitudes, speeds, and background noise levels. The proposed acoustic detectors have a maximum detection range of 150 meters. However, this is an indispensable solution for small strategic areas and borders between countries. And to expand the working area of the sensor, there is a solution with a repetition of the sensor, which is cost-effective [6, p. 138669-138680; 7, p. 3856-2-17]. Acoustic methods have a great advantage in daytime and nighttime operations because they are independent of lighting conditions. Being able to recognize drone states when they have extra payload is another great feature.

The RF approach is another way to identify and classify drones. Based on their RF characteristics, drones can be identified and categorized. In addition, the SDR approach is gaining popularity in this field. The RF sensor, which recognizes radio frequencies, serves as a conduit between the UAV and its controller. When listening to UAV controller signals, RF sensors, unlike acoustic sensors, overcome the issue of limited detection range by using high-gain receiving antennas in conjunction with high-sensitivity receiving systems [25]. The issue of environmental noise is also addressed by using some noise cancellation techniques, such as bandpassing [7, p. 3856-2]. In the case of detection of drones without RF transmission, low-cost camera sensors based on computer vision algorithms and acoustic sensors can be used. Drones can be visually detected using camera photos of the scene, and these approaches are easy for humans to understand, have acceptable localization, average detection range, and reasonable cost. However, this method does not work well at night or when visibility is poor due to clouds, frost, or pollution. The use of thermal imaging cameras can be a solution to some of these problems. However, for military purposes, high-quality thermal imaging cameras are used. Available commercial thermal imaging cameras may weaken in high humidity or other adverse environmental conditions [6, p. 138675-138676; 7, p. 3856-3-4; 20, p. 26-1-26-5]. Each of the areas discussed above has its own successful recognition skills for certain areas of interest. In accordance with this, the scope of the method is selected. And in this work, we are considering a solution to the problem of incidents or the penetration of drones with special cargo, which is considered very dangerous for the life of all mankind. The focus of the problem is not the territory and not the range of protected areas, but the state of drones penetrating the territory. As we discussed above, the acoustic recognition method is an effective solution for recognizing and classifying the states of these small UAVs. And in the following subsections this method will be widely discussed, and technical solutions will be sought.

1.2 Acoustic data-based UAV detection

Nowadays, the problem of sound classification of UAVs has aroused particular interest in the scientific community due to their ability to detect UAV states in the presence of an additional load on them, at different positions or models [1, p. 863; 15, p. 453-454; 16, p. 470-473; 20, p. 26-2-26-4]. Thus, this work is aimed at studying the problem of UAV sound recognition. Moreover, the importance of this method increases due to its ability to estimate distances from the interest areas. The detection

of Unmanned Aerial Vehicles (UAVs) in protected areas using acoustic signals expands the capabilities of accurate detection of harmful UAVs for timely activation of the security system. UAVs are now more widely available due to their affordable price and many features, which have increased their use in terrorist and criminal strikes. Furthermore, technology has advanced, drone design and development have become more affordable, and their potential applications have grown quickly. Drones are especially utilized for the purposes such as pesticide delivery, food delivery, search and rescue operations, transporting flocks of birds toward airports, small spy drones, disaster relief, and agriculture, and the list is continually expanding [3, p. 149; 20, p. 26-1-26-3]. Along with the numerous uses of drones in the airspace, security concerns are associated with them, such as endangering the airspace itself, invasion of privacy, use of vehicles as weapons, corporate espionage, vehicle collisions, and drone hacking. One such incident occurred in which terrorists killed two soldiers while smuggling explosives using UAVs in October 2016. Prolonged use of such weapons can lead to mass casualties in metropolitan areas, where it is easy to hit a large number of people [6, p. 138670-138675]. On September 29, 2022, a different incident happened when an Alphabet subsidiary Wing delivery drone collided with power lines in the Australian city of Browns Plains, knocking off electricity for almost 2,000 clients. An unknown drone payload has stalled on an overhead power wire. In this case, even removing the drone from the cable turned out to be impossible. Although it did not turn off the power, the drone followed the voltage until it landed at 11,000 volts, burst into flames, and crashed to the ground. 2,000 local residents were left without electricity for about 45 minutes, and another 300 remained without electricity for three hours so that power engineers could check the lines for damage [27], figure1a. A number of Amazon's recent drone crashes have also been caused by engine and propeller problems [28, 29], (figure1b).



a – A loaded drone in the city area; d –A loaded drone on a path of power lines

Figure 1 – Delivery drone crash into power lines

As we can see above, there has been an increase in cases where unmanned aerial vehicles (UAVs) have been widely used in hostilities in recent years. The work [29] provides an extended list of incidents with drones in the military and other different situations. Generally, non-military UAVs have often been implicated in incidents where they have endangered aircraft as well as people or property on the

ground. Because a swallowed drone can quickly damage an aircraft's engine, there have been safety concerns. Several confirmed collisions and hazards have involved amateur drone operators, too, who have flown in violation of air safety laws. These views claim that the identification and categorization of UAVs will always be of paramount importance. And the acoustic sensor method can be an effective solution to the problem of drone detection and classification. Due to the advent of multifunctional technologies that have allowed drone users to create their own drones, and the near impossibility of monitoring them, other methods are impractical. The military can identify drones with very sophisticated radar systems, but these systems are expensive, and their practical design is not suitable for urban environments. In addition, there are a number of integrated commercial solutions that use various complex sensor systems such as radar, RF, cameras, and thermal sensors [3, p. 149-160; 6, p. 138682; 7, p. 3856-3-3856-7; 20, p. 26-20-25]. But the drone incidents mentioned above require the definition of models or types, distances of drones to objects, and their loads. And the acoustic sensor method is suitable for the optimal solution of these problems from a technical point of view. That is, if drones are studied by their sound signatures, then it is technically possible to determine their model, state, and position. This is because different drone models have different motors that make different humming sounds, which in turn produce different frequency responses. As a result, enough data can be collected for processing using deep learning methods in artificial intelligence. Also, if the drones are loaded with special mass, even if they are the same model, the sound data will change due to the weight on its engine. Summing up the mentioned factors and possibilities, the study of sound recognition by drones shows that this is an effective solution. The use of deep learning and machine learning methods, which are modern and productive branches of artificial intelligence, is considered the most reliable solution for processing such collected data. The recognition of these objects based on the collection of a sufficiently large number of quantitative patterns and data from the same object with high accuracy can be achieved by training their patterns using neural networks.

1.2.1 The Role of Classification for UAV Acoustic Data Recognition

Activities conducted outside and indoors, whether they include human or non-human activity, almost always contain sound. Technically, sound recognition problem is difficult since the signal is dynamic and complicated, but interest in sound research is growing because it has the potential to develop a variety of applications. Indoor sounds refer to more specialized environments including corporate, residential, and educational settings, whilst outdoor sounds might encompass ambient and urban surroundings. Applications for analyzing human-produced sounds include Automated Speech Recognition (ASR), Speaker Identification (SID), and Music Information Retrieval (MIR). And the sounds produced by moving objects can be called sounds caused by the impact of these objects and vehicle engines. Such transport objects may include cars, types of motorcycles, trains, aircraft and drones. Recognition of the sounds of these objects for security purposes is also one of the

important issues. In order to support sound recognition, machine learning has been utilized, including both "conventional" (classical) techniques and deep learning. And the most key question for sound recognition with these machine learning methods is the processing of sound signals. Signal framing is a component of sound preprocessing. It can be difficult to choose the frame length that will both fully capture the sound of interest and exclude other sound kinds. Thus, traditional machine learning methods and deep learning are the foundation of good sound recognition research. Sound features are extracted and incorporated into standard machine learning algorithms. The features of sounds are based on their acoustic characteristics, such as loudness, pitch, and timbre. Spectrograms, Mel spectrograms, Mel-Frequency Cepstral Coefficients (MFCC) and their derivatives are also often used as cepstral features. The focus of this dissertation work is the recognition of sound differences of objects. And the recognition of sounds coming from common objects has become one of the most effective ways to identify them by classifying them. And in the case of this classification, a number of scientific papers have confirmed that it is possible to achieve good recognition by effectively processing the mentioned acoustic characteristics of the sound signal. Deep learning techniques for sound recognition have recently become the subject of considerable research. Deep learning techniques differ from the conventional application of classical techniques in that in the former, a pool of values from the time, frequency, and perceptual domains are retrieved and fed into ML algorithms. The following chapters will analyze the efficient processing of the acoustic parameters of the sounds of drone objects and the research of recognition with deep learning.

1.3 Related works on UAV sound detection and classification methods

Generally, unmanned aerial vehicles (UAVs), as was previously said, pose a serious threat to public spaces like parks, schools, hospitals, and government facilities. Different drone-detecting tactics are getting more and more active. Researchers are particularly interested in the detection of UAVs using their acoustic signals since it is less expensive than other conventional options and can solve more scrupulous issues. Therefore, specific researchers in this field believe that using an acoustic signal print to investigate a drone detection system is an effective technique [30, 31]. This section focuses mostly on investigating cutting-edge scientific methods for recognizing drones by their distinctive acoustic signal fingerprints. That is, a review of the literature is given on various analyzes of already used drone sound detection systems that are being studied and used to solve the complex problem of determining the speed and unstable state of these small-looking vehicles. As we are all aware, the four key areas where the drone detection system is being developed are radar, computer vision, acoustic sensors, and radio frequency sensor systems. Organizations in charge of air traffic control are working hard to lessen the risk posed by drones. The use of drones that is irresponsible or untrained may be stopped by the current restrictions, but attacks by criminals or terrorists will still happen [6, p. 138670-138673]. Accordingly, scientists are intensively searching for an effective method of "Drone Detection System" that meets the requirements of emerging tasks

and prevents attacks of all types of threats. High performance results are being produced by efficient systems that fulfill these needs and objectives while using machine learning techniques, which are impacted by the quick advancement of contemporary research. Additionally, it has developed into an intriguing, complicated, and difficult topic of technology to solve utilizing Deep Learning, a subset of Machine Learning that is being investigated extensively. Most of the research relies on standard machine learning methods (including KNN - K Nearest Neighbor, SVM - Support Vector Machine, Random Forest, etc.), while the most recent research demonstrates the need for deep learning methods. In order to respond to the scientific inquiry regarding potential threats, this work has only selected and examined one particular direction from drone detecting systems. In other words, the central scientific issue is the detection system required for drones in their most hazardous situations, such as when they are flying in specific locations with extra load or in specific positions. Within the framework of this question, it was considered that among the four methods of drone recognition, acoustic sensors are the most effective method. It is also assumed that this method is effective for the operation of the system regardless of environmental factors in the task of detecting dangerous drones [31, p. 1-2]. According to the development of study in recent scientific works, the technological direction of the acoustic method is split into two main distinct regions: Drone detection and localization [6, p. 138677; 31, p. 302-303], (figure 2).

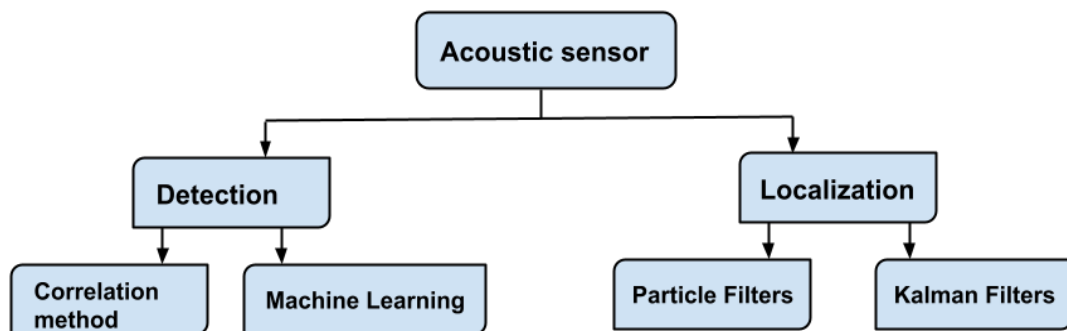


Figure 2 – Directions of the acoustic sensor method for UAV detection

Additionally, (figure)1 in work [5, p. 2-3] classifies and illustrates the methodologies used by these two branches. This section provides an overview of acoustic approaches, including a thorough examination of the detecting field within the context of machine learning and deep learning, in contrast to the research work [6, p. 138674-138675], which gives a comprehensive overview of all forms of drone detection systems (hereinafter, if "detection" or "localization" is considered, that these are methods based on acoustic sensors). The following primary goals are distinguished by techniques like machine learning or deep learning for a comprehensive solution to the drone detection problem: a - a binary classification, indicating whether a drone is present in a specific area; B. Multiple classification, including loaded and unloaded, with transport of goods (for instance, cargo 0.5 kg, box, damaged box), without transport of goods, or classifications for various drone

models. Analyzes of scientific papers related to acoustic sensors based on machine learning and deep learning were carried out (table 1). And research on individual methods of each direction is discussed in the following sections.

Table 1 – Analyzes of scientific works related to acoustic sensors

HMM, GMM; Machine learning SVM	Deep Learning CNN	Deep Learning RNN	DL: RNN - CNN	DL: CNN-RNN
	UAV load estimation with CNN			
	Multispecies classification with CNN: distinguishing unloaded and loaded drones			
	Multiple classification with CNN			
Classification with PIL, KNN				
		Multiple classification		
			Multiple classification	
Binary Classification with GMM	Binary Classification with CNN			
	Binary Classification with CNN	Binary Classification with RNN	Binary Classification with CRNN	
Multiple Classification with HMM				
Classification with Random Forest				
Classification with SVM				
Complex methos on Classification with SVM				
Note - Compiled according to the source [1, p. 863-865; 2, p. 241-244; 3, p. 3-148; 4, p. 2-4; 15, p. 452-457; 16, p. 471-473; 26, p. 1858-1860; 32-36]				

The literature review focused on research done with acoustic sensors using machine learning and deep learning methods. The results of the discussion proved, as can be seen from the table, that Deep Learning methods provide high performance, and among them only the CNN network has been studied more. And very little research has been done by other authors on RNNs with binary classification. A

scientific study was carried out on the LSTM network, which is a type of RNN network, and published [15, p. 456-457; 16, p. 473]. From this we can see that a complete study of RNNs by other authors has not been carried out, although RNNs have been successfully used for audio signals. However, drone sounds must go through preparatory processes that can be taken into account during training in order to apply these algorithms. Namely, the acoustic structure of the UAV detection system consists of the following main parts: *data preparation*, *preprocessing* and *classification*. Data preparation is associated with the collection of acoustic data from various types of UAVs using acoustic sensors, i.e., microphones. Pre-processing considers getting ready audio data to the Network by extracting features from audio representations. The classification task concerns training datasets using machine learning or deep learning methods [15, p. 454]. And the following sections discuss literature reviews related to these parts: pre-processing methods that prepare input acoustic data for neural networks, machine learning, and deep learning methods.

1.3.1 Pre-processing methods of the UAV acoustic data recognition system

A branch of computational sound analysis called classification of environmental or objects sound events aims to build intelligent machine listening systems that can recognize acoustic situations that are familiar to human listeners [3, p. 3-145]. And intelligent machine listening systems require technically to perceive these sounds in a special format, which is called pre-processing of audio signals. The pre-processing phase is extracts features from the UAV's acoustic representations. That is, acoustic data must go through a number of pre-processing procedures before analysis, just like any other unstructured data type. Basically, to analyze sounds, two different types of features are used: Time domain features and frequency domain features. Most studies have shown that time domain analysis is insufficient for machine learning. As a result, processing in the frequency domain has become more widely considered. The amount of processing space needed is greatly reduced when data is provided in the frequency domain. As a result, the sound signal is divided into different pure signals, each of which can be represented by a different value in the frequency domain [38]. Basic operations from Fourier transform analysis, continuing with spectrograms, filterbank coefficients, Melspectrograms, and MFCC, are used to obtain frequency features [6, p. 138678; 15, p. 454; 16, p. 473; 18, p. 127; 39]. Figure 2 [6, p. 138677] shows a visual representation of these categories of audio characteristics that can be extracted for analysis as a whole. Due to the complexity of the signal, processing using spectrograms, filterbank coefficients, Melspectrograms, MFCC frequency domain approaches, or other extra filters is also taken into consideration in many research.

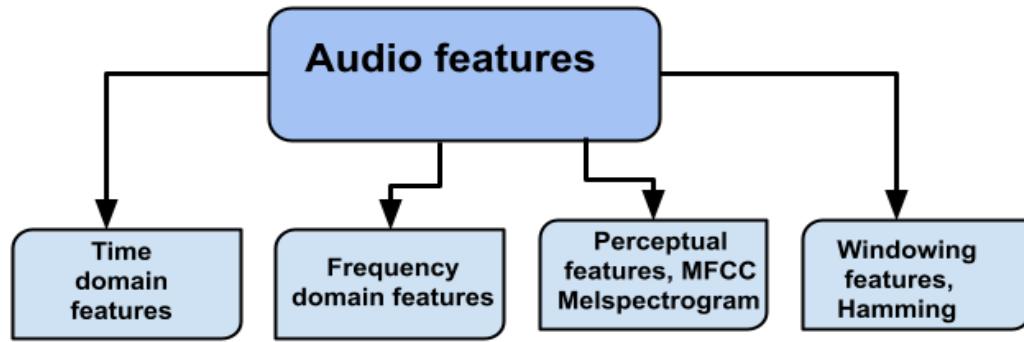


Figure 3 – Types of features during preprocessing of audio signals

As a rule, processing of signals in the time domain is mandatory, since the entire range of information domain of signals can be explored in the time domain. Since it is difficult to process audio signals recorded by microphones, after defining the information domain in the time series, the frequency domain is further considered.

Since the audio signal is continually changing over time, these components are presumed to be constant if the area value is obtained over a short period of time, which is another reason to take into account the frequency domain. A short-time Fourier transform (STFT) was used to retrieve these constant portions. The linear frequencies are then transformed into logarithmic frequencies and analyzed using the Mel scale. A filter serves as the basis for filter banks. The filter bank coefficients are protected from energy decorrelation using the discrete cosine transform (DCT). It compresses all information to lower frequencies. MFCC coefficients (Mel Cepstral Coefficients) are obtained as a result. Melspectrograms can be acquired as an alternative [40, 41]. Numerous studies in this area have demonstrated superior MFCC frequency response performance for audio classifiers. MFCC provides useful qualities, especially for capturing the periodicity of the fundamental frequencies produced by the drone's rotor blades. The authors of [26, p. 1859-1860] captured and displayed the sound of a drone using MFCC functions in an audible harmonic structure. The authors also noted a distinct area of effect on the spectrogram between 5000 Hz and 7000 Hz. But not every drone model used in their tests included these features. In terms of energy, low frequency data travels farther than high frequency data. Because of this, the authors only examined low frequency data below 1500 Hz. The smallest audio data length converted to an MFCC vector that performed optimally with the GMM configuration was 40ms with 50% overlap. When processing data of at least 240 ms length converted to a Melspectrogram with a Melbin of 40, the other models, CNN and RNN, have demonstrated the best performance as 80.09%. While the task can identify drone sounds, it is constrained by its inability to provide a high accuracy score and by not considering the drone's state, such as its load. The authors of the work [1, p. 864-865; 42, 43] investigated and used the extraction of MFCC and STFT functions from an optimized system with several acoustic nodes. In [33, p. 510-515], a method for recognizing drones by the sounds made by their propellers was proposed. This method used the Mel frequency cepstral

coefficient (MFCC) method for feature extraction for classification. The first of two feature extraction algorithms, which also adds dynamic features, uses twenty-four MFCCs and a forecasted thirty-six MFCCs. And the authors in [6, p.188677; 18, p. 127-129] used a variety of feature representations based on signal processing methods, comprising pitch, energy, zero-crossing rate, mean-crossing rate, spectrogram, Mel-frequency cepstral coefficients (MFCC), Mel logarithmic spectrum, and Spectrogram extensively.

1.3.2 Machine Learning algorithms for UAV acoustic data recognition

A drone in flight makes a buzzing noise that acoustic sensor may record and analyze using various techniques to determine unmanned aerial vehicles sound pattern. Machine learning classification or correlation/autocorrelation approaches are used for acoustic drone recognition in the initial studies in this direction [6, p. 188677-188678]. A feasibility study for drone detection from sound was given by Nijim and Mantrawadi [6, p. 188670-188675; 44]. To find the DJI Phantom 3 and FPV, they used a Hidden Markov Model. To identify a drone within 150 meters, Jeon et al. suggested utilizing the Gaussian Mixture Model (GMM), CNN, and RNN classification [26, p. 1859-1860]. By combining various environmental noises with uav sounds, the authors proposed constructing datasets in order to solve the paucity of audio features for flying drones. Using various uavs to train and evaluate the classifiers seems to be an intriguing component of their research. They discovered that the RNN classifier outperformed the others (80%), GMM (68%) and CNN (58%), respectively. With unknown data, although, all classifiers perform noticeably worse. To distinguish the drone noise from other signals like crowd and daily nature sounds, Bernardini et al. employed a multi class SVM classifier [6, p. 188672-188675; 37, p. 61-63]. To use an audio file extractor, the task involves gathering web sound data with an emphasis on files with sampling rates greater than 48 kHz. The dataset comprised five 70-min UAV flight sounds, together with audio from daylight nature, busy streets, passing trains, and crowds. Then, the acquired data were divided to overlapping 10-ms chunks lasting 5 seconds for midterm analysis and 20-msec subframes for short-term investigation. In order to train a Classification model, the authors have provided characteristics from pre-processed data, including short time energy, temporal centroid, Zero Crossing Rate (ZCR), spectral centroid, spectral roll-off, and Mel – Frequency cepstrum Coefficients (MFCCs). 96,4% accuracy was achieved when comparing the results for detecting the drone sound to the other categories. In order to identify the DJI Phantom 1 and 2, Kim et al. [39, p. 544-547] suggested employing correlations, spectrum representations of sounds, and k-nearest neighbor (KNN) classifier techniques. Various audio signals have been captured from the UAVs inside (without propeller) and outdoors, as well as from a drone-free outside area and background noise from a YouTube video. Each recorded sound was divided into frames of one second in this work. Image correlation produced accuracy of 83%, and KNN produced accuracy of 61%. An acoustic wireless sensor network (WSN) with machine learning (ML) was used by Yue et al. to construct a distributed system that could detect UAVs and determine their approximate location [45]. The

scientists conducted a number of studies and discovered that the unmanned aerial vehicles sound's PSD differs from other ambient noises. After preprocessing an unmanned aerial vehicles sound using a lowpass filter (LPF) with a cutoff frequency of 15 kHz, the PSD is derived using the Fast Fourier Transform (FFT). The outcomes of the study demonstrated that the audio signal may be filtered to remove unwanted noises at around this cutoff frequency. A PCA-based dimension reduction method was used to train an SVM classifier to distinguish the drone sound from other sounds (rain, ambient). The database was compiled from many classes, each of which has 20,000 samples. Subsequently, 2000 tuples have been arbitrarily chosen and divided into 50, 30 and 20%, that were utilized for validation, training, and producing overlapping signals for subsequent testing. For the test case with a signal to interference ratio (SIR) greater than 10dB, supplementary Gaussian noise has been applied. The results show that this amount of injected SIR or higher was effective in detecting the drones. Seo et al. suggested using the normalized STFT to extract 2D pictures from the audio signal of drones [6, p. 138671-138678; 46]. First, 20-ms segments with 50% overlap were created from the acoustic source. A method with numerous calibrated acoustic nodes was suggested by authors of [47] to extract the MFCCs and the STFT characteristics. The characteristics were then used to train the SVM and CNN classifiers, two types of supervised classifiers. The CNN structure provided for the representation of the sound signal using 2D images. This model included pooling and dropout layers, as well as two convolutional layers, two FC layers, and four full layers. During the initial instance, the drone had a maximum range of 20 meters and also was flying between 0 and 10 meters above the 6-node acoustic apparatus. In another instance, data collection took place without the use of an UAV, and the only sound recorded was background noise. The Parrot AR Drone 2.0 was the one of the UAV models examined [6, p. 138678].

1.3.3 Deep learning algorithms for acoustic data recognition

The possibility of sound understanding similar to human could lead to a wide range of applications, such as intelligent machine state monitoring, using acoustic information, acoustic surveillance, categorizing, and information extraction applications. Environmental sounds are more varied and cover a wider frequency range than speech [18, p. 127-128; 19, p. 456-458]. And the majority of the existing research on sound recognition relies on conventional classifiers like GMM and machine learning methods, which lack the feature abstraction capabilities present in deeper models. And in works such as [18, p. 128], which extensively studied audio signals, the results of successful analysis of the frequency content were achieved using deep learning algorithms.

The authors concentrated on the problem of classifying an acoustic scene, which involved selecting a semantic label to describe the acoustic surroundings of an audio stream. Modern deep learning (DL) architectures have been applied to a variety of feature representations generated using signal processing methods. They utilized the following designs, specifically: 1. Deep Neural Network (DNN). 2. Recurrent Neural Network (RNN). and 3. Convolutional Deep Neural Network (CNN). The

following models were also investigated in combination: (DNN, RNN and CNN). Along with i-vectors and the mixed Gaussian model (GMM), we also contrast DL models. The authors found that deep learning models compare favorably with traditional pipelines (GMM and i-vector). In particular, GMM with MFCC function, the base model achieved a test accuracy of 77.2%, while the best performing model, which is a hierarchical DNN, achieved a test accuracy of 88.2%. RNNs and CNNs typically have performance in the 73-82% range. Combining temporary specialized models such as CNNs and RNNs with specialized resolution models (DNNs, i-vectors) greatly improves overall performance. The fact that the most efficient model is a non-temporal deep neural network model suggests that environmental sounds do not necessarily exhibit strong temporal dynamics. This is consistent with our everyday experience that environmental sounds tend to be random and unpredictable [18, p. 127-129].

Recent research into drone sounds is ongoing in deep learning networks. This is because the results of CNN networks in this area were with high recognition accuracy. One of the successfully studied work in the field of UAV sound recognition [26, p. 1859-1861] showed the best performance as 80.09% with the use of CNN and RNN algorithms. Their research looked into how a deep neural network may be used to identify commercial hobby drones in actual environments by examining their acoustic data. Their work was intended to aid in the detection of drones used for nefarious purposes, like terrorism. In specifically, they made an effort to outline a technique for detecting the existence of commercial hobby drones as a binary classification problem based on the detection of sound events. To cover the gap in drone sound data in diverse contexts, they recorded the sounds made by a number of well-known commercial hobby drones. They then supplemented this data with a range of environmental sound data. The effectiveness of these models was evaluated by empirical research on a test dataset gathered from a city street. Their RNN models thus had the best detection performance, with an F-score of 0.8009 at 240 ms of input audio and quick processing times, demonstrating their suitability for real-time detection systems. However, this work did not undertake the study of drone states. The study [32, p. 460-463] attempted to offer approaches for identifying and detecting drones utilizing deep learning tools as convolutional neural networks, recurrent neural networks, and convolutional recurrent neural networks (CRNN). In order to locate and identify flying drones, these algorithms took advantage of their distinctive acoustic signatures. Based on their dataset, which contains audio recordings of drone movements, they suggest comparing the effectiveness of various neural networks. The main contribution of their work is to give a robust evaluation of the performance of various deep neural network algorithms for this application and to confirm the utilization of these approaches of drone detection and identification in real-world scenarios. The paper attempted to demonstrate the ability to identify drone models with a maximum recognition rate of 93.83%, but did not address the study of other suspicious drone positions. As can be seen from the works discussed, most of the advanced work on the study of sound recognition by drones has been carried out using deep learning neural networks, and in the works on binary classification or

classification of several UAV models, networks with CNNs have been carried out using high performance. However, these works did not represent an extended task. And one of the successful and large-scale works in this direction is the study of the RNN network with the highest score among other algorithms [32, p. 463-464]. But this work did not take into account the state of the UAV. As a result of all the ongoing work, the widely studied deep learning algorithms have evolved as shown in figure 4.

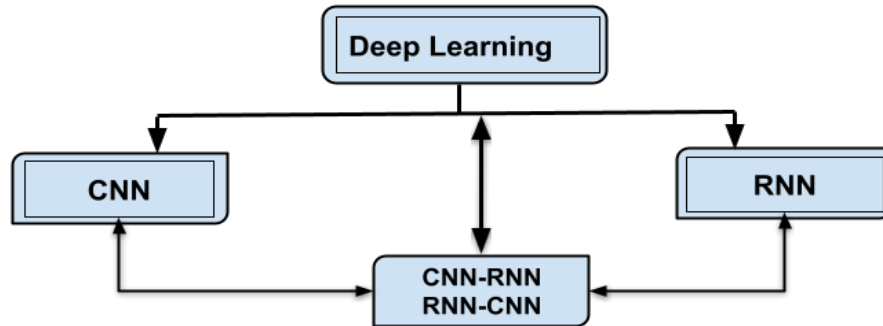


Figure 4 – The most studied networks in deep learning for UAV detection

During the research phase of this dissertation, an experiment was conducted using a CNN network and published based on that experimental work. When studying publications [1, p. 865-866; 2, p. 244], which are analogues of the research of this dissertation, new CNN network structures were processed, adapted to the newly created database, table 1 in Appendix A. When developing layers in the model structure (CNN) used in the publication [48], a final BatchNormalization layer was added to the fuzzy layers as a method of stabilizing the operation of artificial neural networks and improving their performance. Since the project is based on 3 classes, the softmax function and the filter size of 3 were used. And the “sparse_categorical_crossentropy” function was taken as the system loss function. In the experiment of the research paper [48, c. 42-43], 2 different databases were considered. The first database was compiled based on the complex background noises of the drone sounds of the DJI Phantom 2 system. The DJI Phantom 2 model is widely used to detect payload drones and is a vehicle capable of carrying 0.5 kg of payload compared to the DJI drone Phantom 1. When recording sounds, situations with complex sounds (wind, motorcycle, train and cars) were considered. Consideration in such a complex situation is the main focus of this study. This is because building a robust system for real-world applications requires separating drone flights from other superimposed complex noises. The second base is supplemented with the sounds of the DJI Phantom 1, 2, 3, 4 drone models, as well as the acoustic signal data of the Syma x5, x20, Tarantula drones, which are widely used as children's toys, and other types of drones in open areas. These databases have been validated against two different CNN model structures set out in table 1 in the Applications section. In general, both databases were divided into three parts: 60% training, 40% verification and 40% testing. The data in the test and validation parts were not pre-trained by the model. The results of training and testing data using

model estimation (model estimation) are shown in figures 5, 6, and test data using model prediction.

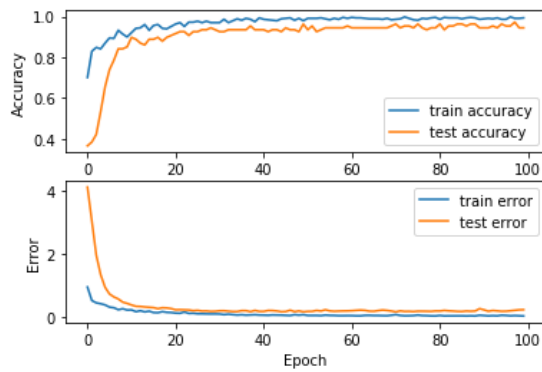


Figure 5 – Accuracy and loss models for the initial database of training using CNN

This demonstrates that as the database grows, the neural network recognition system's identification accuracy has an optimal performance score and the ability to train the model, as shown by the second graph.

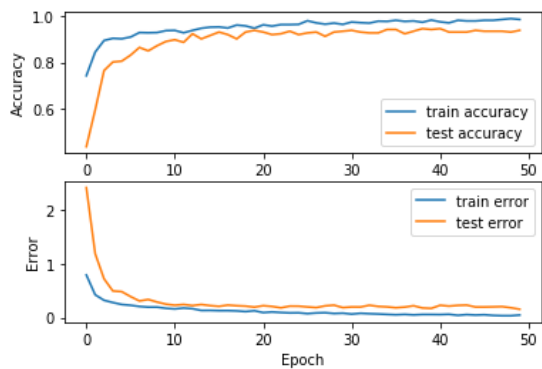


Figure 6 – Accuracy and loss models for the modified database of training using CNN

At this stage, a feature vector of randomly selected items in the test data, i.e. audio image data values, is presented in table 2 to accurately represent the prediction results. The main goal is to test the system created as in works [1, p. 863; 2, p. 243] against a database of many different drone models, to study its reliability in a real-time system, and to identify the main problems.

Table 2 – The results of the study of CNN models in the publication

Dataset	MFCC size	Models	Accuracy
Dataset I (DJI P2)	(63, 20, 1)	CNN by [1]	94.41% (100 epoch)
		CNN by [2]	89% (20 epoch)
Dataset II (Many models of UAVs)	(63, 20, 1)	CNN by [1]	96.1% (50 epoch)
		CNN by [2]	81% (30 epoch)
Note – Compiled according to the source [48, p. 41-42]			

The study in this publication required the addition of several layers of CNN work and many more epochs, using data from a newly collected database to test the new target. This showed that the study with the CNN model is limited by such reasons as a large number of trainable parameters and the use of excess time for training them. Considering the above studies, it was noticeable that with the help of an acoustic signature, it is possible to perform a binary classification of the UAV, as well as determine the load of the drone. In the course of which a research task was carried out on two targets in the form of a publication: the first goal was to create a classification system with many models, and the second goal was to research LSTM one of the types of RNN, that is, the development of a multi-level bidirectional long-term short-term memory (LSTM) with two hidden layers to categorize the sounds of multiple UAVs. The collected data used in the research in [16, p. 471] was gathered for three primary classes of multiple UAV detection, including background noise, the sound of unloaded drones of different models in the scene, and loaded drones in the scene, (figure 7).

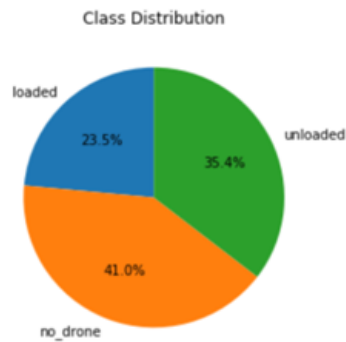


Figure 7 – UAV audio data distribution

Note – Compiled according to the source [16, p. 471]

	fname	label	class
0	P1_stationary_manipulate.wav	Phantom I	unloaded
1	P1_stationary_noise.wav	Phantom I	unloaded
2	P1_stationary_stretch.wav	Phantom I	unloaded
3	P1_up_and_down_manipulate.wav	Phantom I flying up and down	unloaded
4	P1_up_and_down_noise.wav	Phantom I flying up and down	unloaded
...
250	Syma loaded 3_noise.wav	syma	loaded
251	Syma loaded 3_stretch.wav	syma	loaded
252	6 Axis Gyro unloaded 4_manipulate.wav	6 Axis Gyro	unloaded
253	6 Axis Gyro unloaded 4_noise.wav	6 Axis Gyro	unloaded
254	6 Axis Gyro unloaded 4_stretch.wav	6 Axis Gyro	unloaded

Figure 8 – UAV audio data distribution

Note – Compiled according to the source [15, p. 454]

This research project's primary objective was to create a multi-label classification system. It is a classification challenge for many labels due to the dataset's architecture (figure 8). A frequency of 44100 Hz and a microphone bit depth with a resolution of 16 bits were used to record UAV sounds (DJI phantom I, DJI phantom II, Syma x20, 6 axis Gyro, tarantula, etc.).

Uncompressed WAV files have been used to store the audio recordings. Additionally, the set of data included 3 primary classes for all audio data: "loaded," "unloaded," and "no drone". Modeling clay weighing 0.5 KG is carried as a supplementary payload by "loaded" class UAVs. The "P1" class of UAVs was designated as the "unloaded" class so the testing findings indicated that they are too fragile to support any sizeable load.

This research's secondary objective was to construct a Recurrent neural network. The model has an input layer, four input dense layers wrapped in TimeDistributed layers, two stacked bidirectional LSTM layers, a dropout layer, a flat layer, and a dense layer. Bi-directional layers are used to enclose hidden LSTM layers. The hidden LSTM layer employed 128 memory cells [49-52].

This computation process is completed by multiplying by two by the bidirectional shell, which adds another layer. The "categorical crossentropy" loss function is tailored for the multi-label classification problem in the implementation of the suggested model. The weights are optimized using the "Adam" gradient descent implementation, and then during model training and validation, the classification "accuracy" is determined. According to the model's evaluation, the network's predicted skill on training dataset was 94.02%. The best precision, however, was only attained in epoch 49 with an accuracy of 94.09% in 57 s, which presents a challenge for the creation of real-time systems [15, p. 457]. As a follow-up to the work from [15, p. 457-458], the construction of LSTM architecture with a mixture of Convolutional Neural Networks (CNN) for UAV sound categorization challenges was examined further in [16, p. 471]. As a result, utilizing the LSTM-CNN architecture, the study [16, p. 472] tackled the issue of categorizing a UAV's sound with several labels.

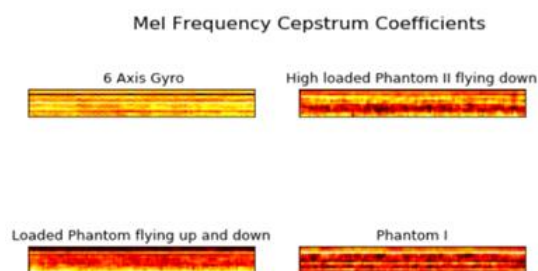


Figure 9 – MFCCs on UAV audio data

Note – Compiled according to the source [16, p. 472]

The suggested architecture consists of CNNs and Stacked Bidirectional LSTMs that have been tested on UAV MFCC representations, (figure 9) [15, p. 472; 16,

p. 455]. According to this experiment's conclusions, employing Stacked BiLSTM and CNN model, Figure 1 in Appendix B, together produced results that were more accurate than utilizing either architecture individually.

However, all studies carried out so far have given high results only on the basis of recognition with many epochs. And the Loaded UAV class had an insufficient database. And the limitations of the works [15, p. 457; 16, p. 473] are that the correct area of the graph does not stop in the history of the recognition graph. As a result, the flow of research continued. And in the next part, an extended statement of the problem of the second study in this direction is considered in order to supplement all the limitations and shortcomings.

1.4 Problem Statement: The protection system for strategic areas from unidentified UAVs based on acoustic recognition

As a result of their many recreational uses, delivery systems, military strikes, reconnaissance, and cross-border political objectives, UAVs are gradually becoming more significant. Additionally, there are terrorist operations and criminal smuggling, including the smuggling of items through borders, restricted locations, and prisons. The issue of drones being used widely and illegally to take pictures or videos from unusual perspectives [1, p. 863; 2, p. 243; 3, p. 149; 6, p. 138670; 7, p. 3856-2] is also brought up. As a consequence, it is critical to identify drones that are loaded. Based on the results of the literature review, it can be concluded that the drone recognition systems investigated thus far have attempted to address the issues of binary classification, classification that distinguishes between various models, and classification that establishes the load of only one model. Further reporting of incidents using drones requires a serious investigation to determine their states, positions, distances in the case of different models in real-time.

1.4.1 Suspicious UAVs with high-risk cases: Loaded and Unloaded UAVs

Cases such as the case of a drone with a load that fell on high-voltage power lines, Amazon drones that flew into a technical fault during transportation, the incident in China of drones that were launched to throw fireworks and fell on people during a holiday, the frequency of suspicious drones that are often launched in various countries for the purpose of military intelligence show that the suspicious cases of drone use are becoming more frequent. And promotes their timely identification in protected areas where human life is important. Their use with harmful loads in the transportation of contraband goods poses a great danger. Drones, which are used as additional weapons or for special reconnaissance, are considered a factor that raises suspicions, especially for densely populated areas and strategic territories. At the same time, drones that are used as an interest or hobby are often loaded with additional power banks for long flights and high-resolution cameras for taking high-quality pictures and videos. In one of these situations, there is a risk of being managed clumsily. Falling into an occupied or hazardous area, whether due to inflexible management, can result in significant losses. As a result of these incidents,

the problem of both loaded and unloaded drone recognition is very important, and in turn, there is a need to develop a system for recognizing both.

1.4.2 UAV distance Identification

Human life is a vital factor that needs to be protected at the highest level and at the right time. That is why there is a need to protect densely populated areas and identify factors that cause suspicion in that area. One of the most dangerous of these suspicious factors is the presence of suspicious drone flights in these areas. And in order to prevent this danger, the issue of timely detection of suspicious drones in the region and its implementation in real time is put forward. And if the drone flight is found in protected areas, the problem of determining how far they are from people, objects and important buildings becomes secondary.

1.4.3 Multiple model UAV recognition

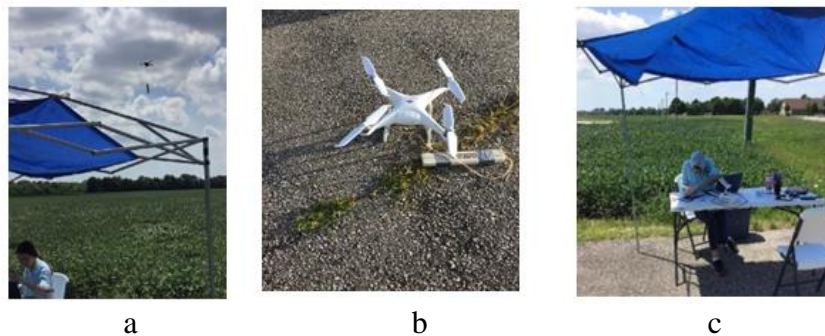
If drones are found flying in protected areas, there is a need to determine their condition or, in some situations, what model they are. In cases where they are found to be suspicious, it is necessary to assess exactly which models have been launched and how much damage they pose. The reason is that UAV models are predictable in terms of how much damage they can cause

2 UAV ACOUSTIC DATA PREPARATION

2.1 UAV sound recording in different positions and models

In the process of researching suspicious UAV actions, it was proposed to create a more reliable detection system for these aircraft based on the recognition of UAV acoustic data, which could identify their sounds in a variety of situations, including their loaded states. The proposed acoustic UAV detection system aims to develop an intelligent audio sensor capable of detecting the appearance of UAVs in certain areas or territories of interest in real time by their sounds if they are flying or if they are flying with an additional load. The next two stages make up the development of this system: preparation of UAV acoustic data for various models and conditions; and architecture structure of the real-time recognition system. In this section, the stage of collecting acoustic data from the UAV is performed, since the initial database must be assigned to start the study.

The initial stage concerned the preparation of acoustic data from the UAVs by recording their flight and their flight with payloads of different models and payload weights, as well as at different distances of 0.5 m and 100 m from the microphone, (figures 10a, 10b, 10c) [20, p. 26-12]. The recording process of the UAV dataset was created by conducting UAV flights with the DJI Phantom 1 and DJI Phantom 2 models with and without payload at a distance of 0.5-100 m from the microphone, (figure 10a).



a – Flight of a loaded drone over a field; b – Loaded drone parking; c – Microphone placement

Figure 10 – UAV audio recording from DJI Phantom series with payload flying over an arable field

The data was collected over several different seasons. A freight train, motorcycles, cars, Gator trucks and background noises with human voices were heard passing nearby while some of the UAVs were launched. Wind, canopy rustling, and other ambient noises were also heard during the testing period, and their data was also collected to distinguish UAV noises from false negatives. During recording, the DJI Phantom 2 was used to launch a loaded UAV carrying a 0.5 kg payload of sculpting clay. The Syma X20 UAV model, which is often used for leisure activities, was delivered using a 0.425 kg metal power pack in both loaded and unloaded situations. When assessing the load on these recreational or amateur drones, the

possibility of harm from a control error from them was taken into account. In addition, additional reasonably priced UAV models have been tested for unloaded UAV enclosures, including Tarantula x6 and Syma x5c ranging from 1 to 40 meters. Other UAV models, including DJI Phantom 1, 2, 3, 4, DJI Phantom 4 Pro, Mavic Pro, and Qazdrone, were also launched with parameters as in table 3 and their noises were added to the dataset.

Table 3 – UAV model specifications and UAV load states

Models of the UAVs	Loading Parameters (kg)	Range (m)
DJI Phantom I	-	2 -100
DJI Phantom II	-	2-100
DJI Phantom II	0.5	unknown
DJI Phantom III	-	unknown
DJI Phantom III	0.454	unknown
DJI Phantom IV	-	2 -100
DJI Phantom IV	0.4	2 -100
DJI Phantom IV Pro	-	unknown
DJI Phantom IV Pro	1.36	unknown
DJI Phantom quadcopter	-	unknown
Mavic Pro	0.156-0.256	unknown
Syma x5	-	1-40
Syma x20	0.425	1-40
Tarantula x6	-	1-40
Qazdrone	-	0.5-30

All of these UAVs were recorded using 16-bit microphone depth resolution at 44,100 Hz, moving up and down, forward and backward at varying speeds depending on their technological characteristics, starting fairly close to the microphone and a nearby parking lot. And the remaining information was gathered from free and open resources. The process of collecting audio files from public sources such as "www.zaplast.com" and "www.sound-ideas.com" required much more effort. This is due to the fact that our prediction system was based on an acoustic sensor capable of listening at a frequency of 44100 Hz and a depth of 16 bits, and the sounds of the loaded UAV were detected only on amateur videos and processed using a special converter at a frequency of 44100 Hz and a bit depth of 16 bits. The rest of the data from open sources were also converted from various formats to a sampling rate of 44000 Hz with 16-bit depth resolution and "mono" microphone mode with the extension ".wav". Since our model was created to receive audio data through the wav extension. The DJI Phantom 2 and its loaded states were the only UAV model considered in earlier studies [1, p. 864-867; 2, p. 243-244] that had this limitation. This study aims to investigate the effect of acoustic data from various UAV models on the problem of complex UAV sounds and their load states. This study aims to investigate how the acoustic data of different UAV models influence UAV load recognition across different models and weights. In general, all UAV recording information was collected and divided into three categories such as "Unloaded", "Loaded" and "Background noise". The three folders include all of these recorded

and collected sounds. Recorded drone noises ranged in duration from a few seconds to more than five minutes. Table 4 provides a general overview of the duration of the sounds collected for each class, in seconds.

Table 4 – Extended UAV sound dataset duration

Classes	Total Duration, in (s)	Train set, in (s)	Prediction Set, in (s)
		7612	7312
Loaded UAV	1513	1413	100
Unloaded UAV	3334	3234	100
Background Noise	2765	2765	100

UAV sounds from open sources included several sounds in "stereo" mode. During the experiment, some of the sounds emitted by Qazdrone, DJI Phantom 2, DJI Phantom 4 and DJI Phantom 4 Pro drones were recorded using microphones from Apple products such as the Apple iPhone 13B iPad AIR 2020. Using a specially created filter, all data files were changed to 44 100 Hz and "mono" mode.

The next two sections are devoted to theoretical solutions for recognizing these initial acoustic database, and the last section explores the development of the system itself and its practical solution.

3 MATHEMATICAL VIEW ON THE SIGNAL PRE-ANALYSIS STEP IN TIME AND FREQUENCY DOMAINS

Generally, sound signal is a complex and non-stationary signal. As mentioned above, more recently, studies on sound classification have become interested in a variety of machine learning (ML) approaches and methods [53], particularly Deep Learning methods. Sound signals are needed to be processed or converted into a format that can be used for machine learning or deep learning so that they can be classified using these approaches. This initial stage or phase in the classification of sound signals using deep learning or machine learning is known as *Signal Preprocessing* or *Feature Extraction*. In fact, all significant characteristics of the sounds of the studied object can be taken into account at the stage of signal preprocessing. Sound signals are first examined in the Time Domain to observe the duration and amplitude of sounds. First, the concept of sound duration determines whether it is simply studied in general or taken into account for real-time systems. Furthermore, an investigation in the frequency domain as an initial source also requires the characterization of time series. Therefore, it should go without saying that every audio signal must first be evaluated in the time domain. And an important reason to study in the frequency range is that a wider range of information can be seen from this range. And if one looks at time and frequency domain together, one can get enough input for machine learning approaches. Such broadly considered signal data has its own specific names for each of their steps and their mathematical foundations. They will be discussed in the following subsections.

3.1 Foundational principle of sound data representation

The study of this thesis examines the processing and recognition of object sounds, in particular UAV sounds, sounds of various motorized objects and the environment. Therefore, this section discusses the presentation of the theoretical fundamental representation of the general audio signal and their representation to pave the way for the study of acoustic signals that can be produced by various objects or backgrounds.

An audio signal is a representation of any sound, often made up of either a series of binary values for digital signals or a changing electrical voltage level for analog signals [54, 55]. In essence, sound occurs when an object's vibrations travel through a medium and strike the eardrum. In physics, sound is a pressure wave. As a result of an object's vibration, the air molecules in its vicinity also vibrate, which creates a series of sound waves to resonate across the medium. Vibration of motorized objects can be generated by the movement of engine components. And UAVs are one of the types of such objects. Sound exists independently of a person's ability to perceive it, whereas the physiological definition also takes into account the subject's ability to hear, (figure 11) [56].

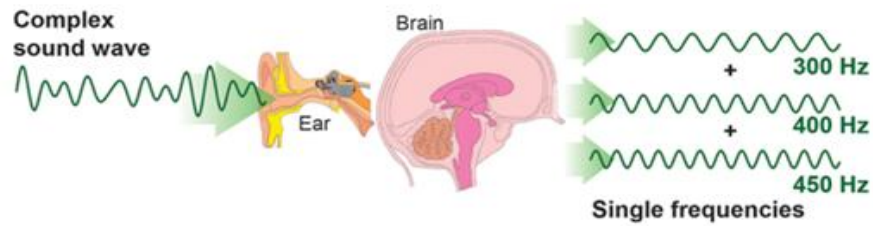


Figure 11 – The process of sound formation and the human perception system

So, a pressure wave is created when an object vibrates, which produces sound. The surrounding medium (air, water, or solid) is subject to a pressure wave that induces vibrational motion in the particles. The sound is transmitted further through the medium as a result of the adjacent particles moving as a result of the particles' vibration. Vibrant air particles cause tiny components of the human ear to vibrate, which causes the ear to detect sound waves [57]. And the trajectory of these particles is similar to a sine waves. In connection with the existence of this physical phenomenon, the study of sound in the form of waves is generally accepted. And sound waves are often simplified to describe sinusoidal waves that have common properties such as frequency, wavelength, amplitude, sound pressure or intensity, etc [58-60].

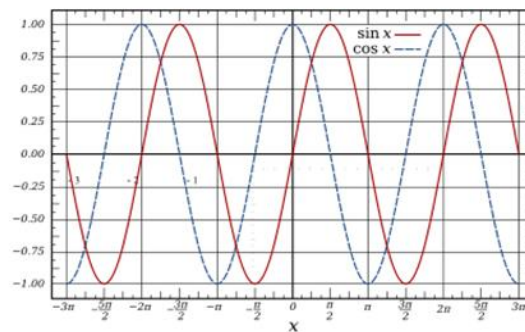


Figure 12 – Sine wave representation

The common formulation of sine wave in Figure 12 can be represented as in the equation (1):

$$y(t) = A\sin(2\pi ft + \varphi) = A \sin(\omega t + \varphi) \quad (1)$$

here A – amplitude, function's maximum deviation from zero;

f – is the frequency, the quantity of variations (cycles) that take place every second of time;

$\omega = 2\pi f$ is the angular frequency, the rate of change in the function's argument, measured in radians per second.

φ is the phase, indicates (in the radians), where in its cycle the oscillation is $t = 0$. When φ is not zero, the whole shape of the wave, apparently, is shifted in time by

the number of ϕ/ω seconds. A negative value is a delay, and a positive value is an advance. Since it keeps its wave shape when combined with another sine wave of the same frequency and arbitrary phase and magnitude, the sine wave is significant in physics. This characteristic is unique to this periodic waveform. This characteristic makes it acoustically distinct and contributes to its significance in Fourier analysis [58; 59; 60].

In general, amplitude and frequency are primary characteristics of sound signals. A signal's magnitude, such as the loudness of an audio signal, is known as its amplitude. Frequency is the number of times per second that a sound pressure wave repeats itself and is measured by Hertz. Since there are a lot of such air particles that vibrate in nature due to adjacent particles, and the mathematical sum of several or many sines or cosines can represent the sounds of certain objects. The addition of several sine waves creates a distinct waveform, which alters the timbre of the sound. Towards the human ear, a sound made up of more than one sine wave will have audible harmonics. Summing up the mentioned phenomena, we can say that sounds are complex signals. Complex signals can be analyzed using discrete time and continuous time models. These two different frameworks for simulating variables that vary over time. A continuous signal is a signal that varies over time and typically has a continuum for its domain. The domain of the function is thus an uncountable set. The actual function does not need to be continuous. A continuous-time signal, also referred to as an analog signal, is a signal with constant amplitude and duration. Every moment of time will have some value for this (a signal). While a discrete-time signal, like the natural numbers, has a countable domain. In the same way that sounds can be presented discrete signals, discrete systems are what will be created for these audio signal processing [59; 60; 61]. The next subsection ensures the study of audio signals in the Time Domain.

3.2 Acoustic Data in Time Domain

Signal processing can occur in any field since auditory signals can be represented electronically in both analog and digital formats. Digital processors operate computationally with binary representations of the signal, whereas analog processors deal directly with electrical signals. The notion of analog and digital signals should therefore be briefly examined. Analog sound is electrical. And as mentioned above, the voltage level is a sound wave of air pressure. On the other hand, the pressure waveform is expressed digitally as binary integers or as a discrete function. In its simplest form, digital representation involves the use of computers and microprocessors. Since digital signal processing techniques are significantly more potent and effective than those based on analog technologies, most contemporary audio systems adopt this strategy despite the fact that analog to digital conversion can be lossy. Sound storage, or the recording and replaying of sounds, is one of the frequently used applications of audio signal processing techniques. Spoken voice, background noise, or music can all be represented digitally by leaving electrical or mechanical traces on a medium, which can then be used to reconstruct the original sound waves. For instance, music was once frequently recorded on CD

players, which we can decode back and record a digital version of the audio signal. Data compression, commonly referred to as audio coding, is another use for signal processing. Reduced storage space needs for audio files and bandwidth constraints for digital audio streams are the main objectives here. There are two different types of compression techniques: lossless, where no information is lost, and lossy, where some information is lost but presumably not any that is crucial for perception. The audio file format which called "WAV files" are one type of perceptual audio coder. There is another audio encoder that can convert the file into a much smaller file. The format of such files is called MP3. This is a very compressed format based on the perceptual characteristics of sounds. Recently, audio formats other than this format are spreading: AAC, Ogg Vorbis, and FLAC [62]. And for signal recognition, audio files are effective in the "WAV" format extension. This is because "WAV" audio files contain lossless information. In the study of this thesis, the "WAV" extension was chosen during the recording of audio signals based on this reason. Accordingly, the first preparatory step is to record the sounds, adhering to these audio file extensions. The data contained in these audio files is considered raw data. The signal processing starts from this raw data. Sounds, as it is known from its physics, are continuous-time signals and change over time. And the processing of audio signals begins with the extraction of discrete-time signals. This process is called signal sampling. Sampling is the conversion of a continuous-time signal into a discrete-time signal in signal processing. The transformation of a sound wave into a series of "samples" is a common example. A sample is a signal's value at a certain moment in time or place. Hence, a sampler is a component or procedure that extracts samples from a continuous stream. A theoretically perfect sampler generates samples at the specified points that are equal to the instantaneous value of the continuous signal. So, it is possible to get a vector stream from these file types by restoring the original signal. That is, the audio signal transmission in the time domain is a long vector [63, 64].

And a time series in mathematics is a collection of data points that have been indexed, listed, or plotted according to time. A time series is often a sequence that is obtained at a series of evenly spaced moments in time. As a result, it is a collection of discrete time data [65]. In audio file signal processing, the first step is frequently to display an audio sample file as time series data. Any recorded audio signal can be displayed in the time domain from the above-mentioned audio files, audio coders or compressors. The representation of any audio signal can be taken as a function $x(t)$. And this function $x(t)$ can be reflected as in figure 13 when they are retrieved from the formats of these audio files, audio coders or compressors during audio signal processing.

Thus, sound can be conceptualized as a one-dimensional vector that holds the numerical values related to each sample. On a time series plot, these sample values can be seen in two dimensions as a function of time $x(t)$, (figure 13).

So, the variation of the amplitude with respect to time can be studied in the Time Series. It forms a long vector containing acoustic information of a given length of time in a time domain.

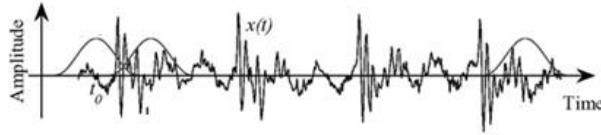


Figure 13 – Representation of the audio signals as a function $x(t)$

There is one strict rule to keep in mind when conceptualizing the audio signal processing. The sampling rate must be calculated using the Nyquist-Shannon theorem when signals with sampled values are considered even in the time domain. The Nyquist-Shannon theorem is a crucial link between continuous-time signals and discrete-time signals in the field of signal processing. It creates a necessary condition for a sample rate that allows a discrete series of samples to fully capture the information from a continuous-time signal with a finite bandwidth. The Nyquist-Shannon sampling theorem offers a requirement for discretizing an analog signal into uniformly spaced samples, making it possible to reconstruct the analog signal from a discrete signal. It also includes removal of aliasing's effect. The process of aliasing blends together several signals. The sampling theorem states that the sampling frequency F_s should be more than twice the maximum frequency component, where f_{max} is the maximum frequency component of the analog signal, equation (2).

$$F_s > 2 f_{max} \quad (2)$$

And in digital devices that have special programs for automatic processing of audio signals, this law is preserved for the sampling rate. It is also necessary to perform signal processing while preserving the conditionality of this Nyquist theorem in programming environments where machine learning is being studied during information recovery from recorded audio file formats and their processing.

Since the time domain only provides information about the amplitude over time, it is not possible to obtain more extensive information. Therefore, by considering them in the frequency range, you can get more information needed for sound recognition. The next subsection will consider the study of audio signals in the Frequency domain.

3.3 Short-Time Fourier-Transform (STFT)

All of the audible sound signals found in nature can be subdivided into a collection of pure sinusoids of various frequencies. A mathematical method known as the Fourier transform uses the decomposition of a signal into its individual pure frequencies to determine the signal's spectral composition. The generated Fourier transform sinusoids for signal as a function of time are a complex value whose imaginary portion is the phase shift of the pure sinusoid and whose absolute value is the value of the corresponding frequency component. And the audio signal is discrete. The discrete Fourier transform, sometimes known as DFT, is the Fourier transform applied to discrete signals. In general, the mathematical basis of the

aforementioned DFT transform, which helps to move from the time domain to the frequency domain, is defined by the following equation (3) below:

$$X_k = \sum_{n=0}^{N-1} x_n e^{\frac{-2\pi i k n}{N}} \quad (3)$$

Discrete signal transformation can be represented by complex numbers and complex trigonometric waves. And the most effective method of calculating the DFT, which allows changing the signal from the time domain to the frequency domain, is the FFT. A Fast Fourier Transform (FFT) is used to represent the signal in the frequency domain and analyze it there, equation (4), [48, p. 39].

$$f(x) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i k x} dx \quad (4)$$

In practice, if a FFT is applied to an audio file of a certain length, it will only provide information about the frequency and magnitude of that audio signal. Processing a signal in frequency and magnitude alone is not enough to create a machine learning data stream. To this end, processing an audio signal while simultaneously storing information about its parameters from the time domain and the frequency domain makes it possible to obtain information about the extended content of that audio signal. The Short-Time Fourier Transform (STFT) helps to carry out this data processing while preserving information from the time and frequency domains. The audio signal is always a changing signal, so we can assume that it does not change significantly during the short intervals in order to simplify the stages of processing. For this reason, dividing the length of the input signal into small time intervals allows you to extract from them a stream of information associated with their frequency. These parts are called frames. Typically, these frames last from 20 to 40 ms and can be created using this input audio signal. If the frame is much longer, the signal is too much fluctuate for the frame, and if it is much shorter, then the samples will not be enough to obtain a reliable spectral assessment. This is carried out on a theoretical basis using the mathematical method of short-term transformation of Fourier, (figure 14). STFT is a Fourier-related transform. It is used to determine the sinusoidal frequencies and phase composition of small signal intervals, since it changes in time. Let function is presented as any audio signal representation, (figure 14). And this function $x(t)$ is performed to divide into a certain "small time segments". Further, the FFT will be calculated for each segment, (figure 14). Small segments in figures 14, 15 are selected by a special rule. This is called Windowing. A "window" in signal processing is a function (shape) that is nonzero for a certain period of time and zero before and after that time. With the exception of the nonzero part of the window, where it exposes the other signal, multiplying it by another signal results in an output of 0. Windowing is most frequently employed in spectrum analysis, which is the process of seeing a brief period of a larger signal and examining its frequency content.

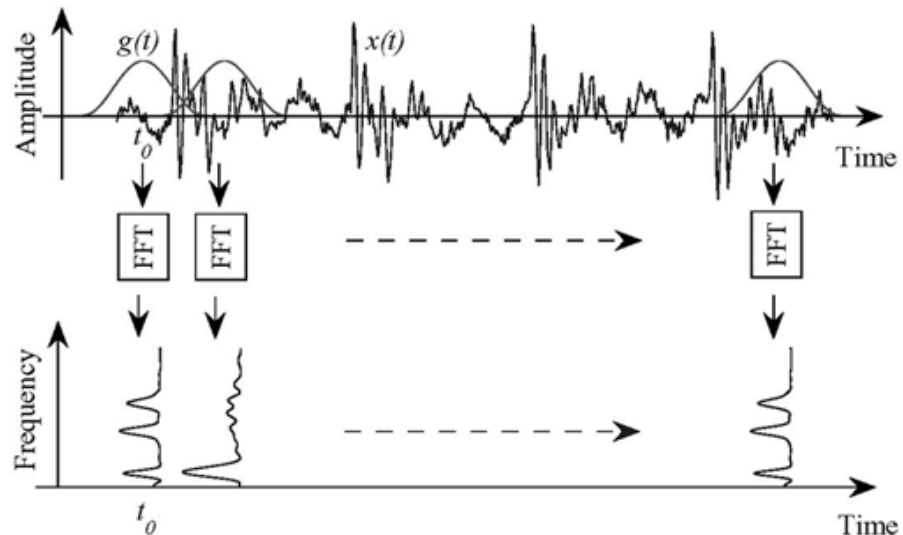


Figure 14 – The fundamental basis of the STFT calculation process for audio signals

Moreover, windows are employed to produce brief sound fragments that last only a few milliseconds. Any finite sound with a beginning and an end can be thought of as a windowed piece of time in general. There are numerous window forms that are possible such trapezoidal, triangular, polynomial windows, and "sine" windows. DFT often employs Hann and Hamming windows [66]. That is, intentional small segments of time or frame are obtained with a certain time. This is called the length of the frame. This frame length adheres to a constantly accepted stable length for all other future segments. The very first frame is taken from the zero point of the coordinate with the length of the frame. As stated above, the frame length is between 20-40 ms. The second frame does not start from the end of the initial frame. The second frame will be calculated with a certain time step, which begins relative to the coordinate of the starting point. This is called the "hop" step.

Typically, the "hop" length is 10 milliseconds. This is also called the step size of the frame. Thus, the next frames are calculated by this rule relative to the previous frames on the basis of sliding until the end of the given signal, (figure 15) in [67]. Figure15 shows that the frames are overlapped during the calculation:

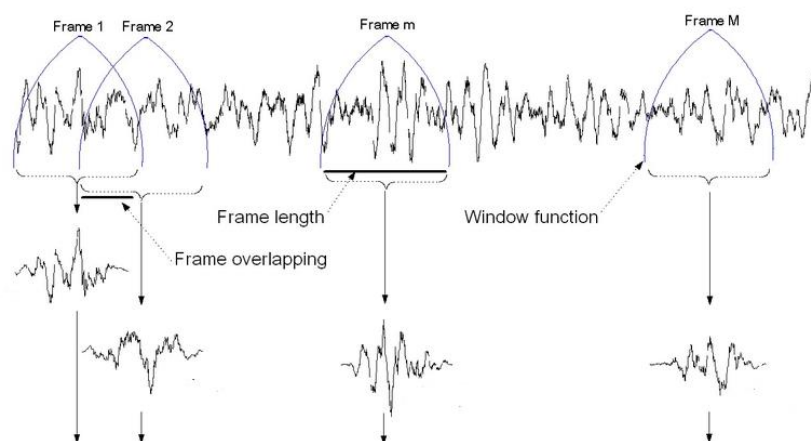


Figure 15 – Visual representation of the calculation of frames

And let's take the length of this window vector as X_i . In other words, X_i is i -th frame of signal x . Thus, the DFT is calculated for each of these received frames according to the sequence. DFT will help to take complex numbers from real numbers, equation (5). As a result, it gives a matrix which has K size:

$$\dot{X}_i \in C^K \quad (5)$$

The discrete Fourier transform is quickly calculated with the formula below as in equation (6):

$$\dot{X}_i(k) = \sum_{n=1}^N X_i(n)g(n)e^{-\frac{j2\pi kn}{N}}, k = 1, \dots, K \quad (6)$$

Here, N is size of the frame signal. And K is the number of FTs to be executed on entire signal. These results are obtained with frequency indicators, that is, they show spectra. The changing spectra are then often plotted as a function of time using a tool called a spectrogram. Values obtained according to formula 6 are complex numbers. Therefore, the absolute values of these complex numbers are obtained. And it gives real numbers, equation (7):

$$P_i(k) = \frac{1}{N} |\dot{X}_i(k)|^2 \quad (7)$$

The results obtained are called periodograms. Thus, the basis of audio signal processing using STFT while preserving the time and frequency indicators of the original audio signal was considered. All these measured quantities are mathematical methods of the signal. And neural deep learning networks work on the basis of the auditory system of the human ear as a part of artificial intelligence. Therefore, it is necessary to process the scale of the studied signal into the logic of the system that the human ear works with. This scale is called the Mel scale [20, p. 13]. The next section deals with the theory of Mel spectrogram processing.

3.4 Mel-Scale Spectrograms

A pitch perception scale in hearing system that is established to be evenly spaced apart from one another is called the Mel scale (after a hearing). The reference point between this scale and the normal frequency measurement is achieved by selecting a perceptual step. The Mel scale is created mathematically from the frequencies. Equation (8) is used to convert from frequency domain to Mel domain.

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \quad (8)$$

In other words, Mel-Scale is a psychoacoustic linear scale representation of frequency. Each of the cochlea's membranes vibrates to a certain frequency component, acting as a crucial bandpass filter in the human hearing system. Stevens, Volkman, and Newmann suggested the "Mel" pitch unit in 1937 as a way to

replicate these characteristics in audio processing [20, p. 11]. Mel is the perceptual scale of pitches that listeners perceive to be equally spaced from one another. Through a series of investigations, it was discovered that the human auditory system perceives signals on a linear scale when they are less than 1000 Hz and on a logarithmic scale when they are over 1000 Hz. Mel-main scale's purpose is to put this characteristic into context. The Mel scale usually produces Mel spectrograms and, in some cases, MFCC coefficient spectrograms. The spectrograms of the MFCC coefficient are calculated using DFT from the values of the Mel spectrograms.

Thus, in this section, the theoretical and mathematical foundations of effective methods of audio data processing for deep learning were given. In the fourth chapter, programmatic and experimental calculations of these operations are carried out.

3.5 An efficient signal processing proposal: the KAPRE method

Acoustic signal recognition systems that use classical deep learning typically pre-process audio signals and store them in separate folders before they are actually trained. These signal processing methods have also evolved in different ways depending on the task at hand and the type of signal. In the previous subsection, these methods were studied theoretically in detail. The STFT, Filter Banks, Mel spectrograms and MFCC were the most efficient methods for processing audio data among other methods. As is known from the theoretical considerations from the second subsection, audio data is processed in two dimensions, starting from STFT, and at the same time, it is known that filter banks, Melspectrograms and MFCC matrices are calculated in stages with the continuation of mathematical calculations. So, the processed audio data, started from the STFT spectrograms, is two-dimensional. That is, they contain both time and frequency data. Each spectrogram, including the STFT, Melspectrogram, and other spectrograms, has one thing in common when examined in further detail: they are all two-dimensional representations of the time and frequency of audio signals. They are helpful because they separate an audio signal, which is simultaneously a mixture of numerous frequency components, into various frequency components. The frequency bins are arranged in such a way that they represent only marginally different frequency components; this gives them a spatial feature. Since the cochlea is used by humans to perceive sound, which also includes frequency breakdown, these spectrograms are based on this idea [43; 68]. As mentioned above, these resulting 2D matrices or spectrogram images were often pre-processed prior to being trained by deep learning methods. This was due only to demand and the search for an effective method among them in the last decade. After confirming the possibility of recognizing sound and voice, their real-time implementation on a practical solution began to gain great demand. However, due to the current demand, their real-time execution and fast processing has become an important scientific demand. The authors of [43] experimentally realized this scientific question on a practical basis.

Currently, the task of recognizing sound signals using deep learning methods is widely implemented using Keras libraries in Python programming environments. Using the Keras libraries, it is possible to process large calculations very quickly

using ready-made neural layers. Data pre-processing frequently requires a significant amount of time and effort, despite the fact that building deep neural network models is becoming simpler with frameworks like Keras that offer pre-built modules. Due to its vastness and intricate decoding computations, dealing with audio data presents more difficulties than dealing with images or texts. Decoding, resampling, and conversion to a time-frequency representation are typically stages included in the preparation process for audio data. Resampling and decoding must be simple to avoid becoming a major bottleneck. There are many options for implementing "time-frequency conversion", each with its own advantages and disadvantages. There is a trade-off between data storage and computation time, whether frequency conversion occurs at runtime in real time or not. Thanks to this, it becomes possible to find the ideal audio pre-processing setting, which is its main advantage. While the decoded audio samples for each configuration often take up the same amount of memory, this can save a significant amount of memory. Therefore, the authors of the work [20, p. 11-12; 43] proposed the KAPRE method, which calculates the time and frequency representations, which is performed as a layer of Keras. And its calculation will be performed when processing on-CPU or on-GPU. One of the key arguments in favor of on-GPU audio preprocessing is its simplicity and speed of implementation. A preprocessing layer can be added with just one code line. With multiprocessing, it can be done on the CPU and might even be faster, but an efficient implementation is difficult. The Kapre approach makes the entire training and preparation process easy. Specifically, creating a generator that loads the data, decoding (and maybe resampling) audio files, saving them in binary formats, and adding a Kapre layer to the Keras model's input side. Thus, the mathematical basis of these spectrograms is the same, and computing the audio data with the Keras method will ensure efficient execution by correctly and properly assigning hyperparameters in a line of code as a Keras layer. And the main advantages of this practical solution for calculating these matrices are the ability to calculate faster than traditional solutions, and the ability to implement coherent learning in the form of adding other neural layers to the deep learning flowchart. The Kapre technique experiments were also taken into consideration for this dissertation's research. The fourth part goes into greater into about it.

4 DEEP LEARNING METHODS FOR UAV ACOUSTIC DATA RECOGNITION

Sound and speech recognition using neural networks has a long history. Neural networks, a subfield of machine learning and the basis of deep learning algorithms, are sometimes referred to as artificial neural networks (ANNs) [69]. And the initial studies of sound recognition showed that machine learning methods, such as vector support machines (SVM), the KNN classification algorithm, K-means and random forest algorithm, were studied by a significant pace, as is noticeable from the subsection of the literature review. And recent studies were widely used to ensure effective results with deep learning methods. The concept of the name of deep learning was formed from the thickness of the hidden layers of neural networks. Traditional machine learning techniques are dominated by convolutional neural networks (CNN), deep feedforward neural networks (DFN), and recurrent neural networks (RNN) in difficult forecast problems [20, p. 2-3].

In recent years, they have attracted attention with the significant improvements in acoustic recognition systems provided by deep feedforward networks. Given that sound is inherently a dynamic process, it seems natural to consider recurrent neural networks (RNNs) as an effective model. In neural networks, recurrent neural networks, are effective models for sequential data [70]. Due to the consecutive occurrence of its connected data points, an audio waveform is a sort of sequential data. Recurrent Neural Networks (RNNs) are able to learn characteristics and long-term dependencies on sequences and data over time. If we conduct a comparative analysis, CNN's ability to learn sequential dependencies has allowed them to gain popularity in applications such as audio processing [42, p. 412-414], speech recognition, machine vision, and image, and video captioning. However, audio signals are constantly changed over time. The consistent and time-varying nature of sounds makes the RNN networks an ideal model for studying the features. Since a RNN has a recurrent hidden state, whose activation at each step depends on that of the preceding phase, it can handle consecutive inputs, unlike a feedforward neural network [49].

Taking into account the factors discussed above, the study aims to explore RNNs in more depth in this thesis. Before studying the RNN network, it was also planned to consider the recognition of drone data using the CNN network for comparison in a practical basis. The results provided by the CNN architecture, explored by previous research work in this area, were compared with the study of RNN network architectures. However, the fact that theoretical predictions and theoretical knowledge about the recognition of audio signals presupposed the effectiveness of the RNN network in advance. So, this section briefly outlines the theoretical foundations of CNNs and provides a detailed mathematical description of RNN network architectures.

4.1 Convolutional Neural Networks (CNNs) in Sound Recognition Problems

Convolutional Neural Networks (CNN) are one of the Deep Learning networks used in various fields such as Object Recognition, Computer Vision, Audio Recognition and natural language processing (NLP) [31, p. 302]. The primary structural characteristic of a CNN is the presence of a standard neural network, which consists of a sampling layer and numerous convolutional layers. Convolutional neural networks are mostly developed for two-dimensional feature-based image recognition. Its input can employ feature layering to accomplish learning and presentation using 2D images. So, there can be many layers in a convolutional neural network, and each layer will learn to recognize different aspects of the image. Each training image is subjected to filters at various resolutions, and the result of each convolved image is utilized as the input to the following layer. Beginning with relatively basic properties like brightness and borders, the filters can get more complicated until they reach characteristics that specifically identify the object. It is very capable of learning, requires little signal processing, and has been used successfully for handwriting recognition, object recognition, face recognition, and sound recognition [71].

A CNN architecture comprises of three layers: an input layer, a group of hidden layers, and an output layer, (figure 16) [72, 73]. It has the three most common layers: convolution, activation, and pooling. The foundational component of the CNN is the convolution layer. It carries the majority of the computational load on the network. With convolution, a series of convolutional filters are applied to the input images, each of which activates different aspects of the images. The next layer type is activation. With the matching of negative data to zero and the preservation of positive values, activation enables quicker and more effective training. “Relu”, “sigmoid”, “softmax” and “tanh” functions are the most popular types. Mostly, Activation function “Relu” is accompanied by Convolution in CNNs (figure 16). So, any intermediary layers in a feed-forward neural network are known as hidden layers because the final convolution and activation function conceal their inputs and outputs. The convolutional layers in a convolutional neural network are hidden layers. A convolutional layer extracts the image into a feature map, also known as an activation map. Layers using convolutions transmit their output to the following layer after convolutioning the input. This resembles how a visual cortex neuron would react to a particular stimulus. Every convolutional neuron only processes information for its particular “receptive field”. Although fully connected feedforward neural networks can be used to learn features and classify data, this architecture is typically impractical for larger inputs (for example, high-resolution images), where it would be necessary to use enormous numbers of neurons because each pixel is a significant input feature. As well, regularized weights across fewer parameters help prevent the disappearing gradients and exploding gradients issues that were present during backpropagation in early neural networks.

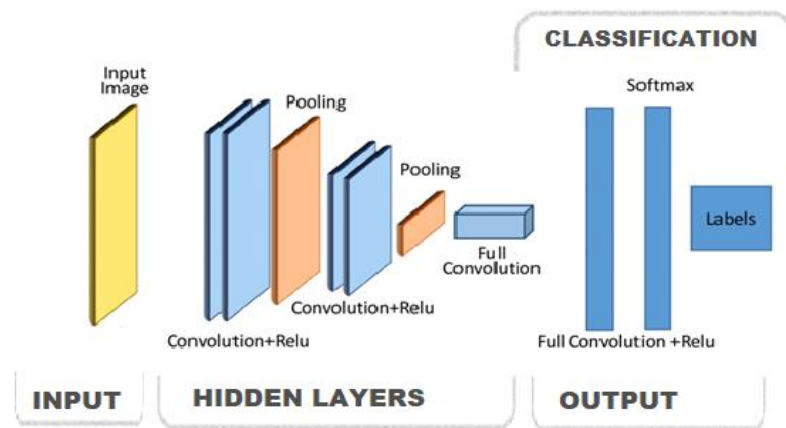


Figure 16 – General architecture of the network

Pooling layer comes next in the list. Using nonlinear downsampling, pooling reduces the amount of parameters the network needs to learn while still simplifying the output. Convolutional networks may also have standard convolutional layers and local or global pooling layers. By merging the outputs of neuron clusters at one layer into a single neuron at the next, a technique known as pooling layers reduces the dimensionality of data. Little clusters are combined using local pooling, which regularly uses tiling sizes of 2x2. Each neuron of the feature map is affected by global pooling. Max and average are the two most widely used types of pooling. When comparing local clusters of neurons in the feature map, max pooling utilizes the largest value whereas average pooling uses the average. The structure of a CNN switches to classification after learning features in numerous layers. The next-to-last layer is a fully connected layer that generates a vector of “N” dimensions (“N” is the maximum number of classes that may be predicted) and contains the possibilities for each class that a target image belongs to. All of the neurons in one layer communicate with all of the neurons in the other layer through fully connected layers. It is equivalent to a conventional multilayer perceptron neural network (MLP). To identify the images, the flattened matrix passes through a layer that is fully connected. The final output of the final classification is provided by a classification layer in the last layer of the CNN architecture [73]. Various types of CNN models have evolved throughout the evolution of the object recognition problem. These include LeNet, AlexNet, ResNet, GoogleNet / Inception, MobileNetV1, ZfNet and Depth based CNNs. And when studying the problem of recognizing sound signals, simple types of convolutional layers created by several layers were used a lot [1, p. 864-866; 3, p. 170; 16, p. 472-473; 17, p. 2-3; 74-78]. The CNN infrastructure is adaptable for image data due to the description of the network and their functionality. Therefore, the next subsection discusses the theoretical foundations of recurrent neural networks, which are considered effective for time-varying signals such as sound [79-83].

4.2 Recurrent Neural Networks (RNNs) in Sound Recognition

The initial and most basic design of an artificial neural network was a feedforward neural network. In this network, data only travels forward from the input nodes, via any hidden nodes present, and onto the output nodes. The network contains no loops or cycles. Feed-forward neural networks for sound recognition tasks have proven attractive in more researches. Moreover, a feedforward network has become popular for solving prediction problems like image recognition, computer vision, speech recognition, sound detection [84-86] and others since it employs multiple hidden layers to maximize learning from the input data [75, p. 229-233; 76, p. 8-10; 77, p. 87-90]. Overfitting is the primary issue with merely utilizing one hidden NN layer. By increasing the number of hidden layers, overfitting can be decreased and generalization can be enhanced. As NNs increase layers, they become Deep FNNs. Deep FF neural networks also have a drawback in that adding more layers exponentially lengthens training time, making FF quite impractical [20, p. 26]. Based on the development of functional shortcomings of feed-forward neural networks, RNN networks have appeared. Recurrent Neural Networks (RNNs) are derived from Feedforward Neural Networks (FF) as a subset. RNNs can extract long-term dependencies and features from sequential and time-series data. The input received by each neuron in an RNN's hidden layers is delayed in time. Current iterations in recurrent neural networks need access to historical data. For instance, one needs to be aware of the words that came before the one they are predicting in a sentence. The RNN can use any lengths and weights as it processes the input over time. This model's computations take into account historical data, and its size is independent of the volume of input data. The slow processing speed of this neural network is a weakness of this network [78]. Based on the solution to this shortcoming, several types of RNNs have emerged. At present, four different computational cells of RNNs such as simple RNN, LSTM, BiLSTM and GRU are popular for prediction. The following subsections provide a theoretical basis for these 4 different RNN networks.

4.2.1 Simple Recurrent Neural Networks (RNNs) in Sound Recognition

RNNs, or Standard Recurrent Neural Networks, are a subclass of neural networks capable of recognizing sequence data. And these networks are widely used by the Python programming environment with the Keras libraries. Standard RNN networks are known as "SimpleRNN" in the Keras libraries. There are three layers in a simple RNN: input, hidden, and output layers, as shown in figure 17. According to Simple RNN's fundamental working theory, nodes are connected to comprehend current information by feeding the output of the neural network layer at time t to an input from the identical network level at time $t + 1$. A series of vectors over time t , such as $\dots, x_{t-1}, x_t, x_{t+1}, \dots$ make up the input data. Input blocks in a Simple RNN with complete connectivity communicate to hidden blocks in a hidden layer. The hidden units in the hidden layer are as follows: $h_t = h_{t-1}, h_t, h_{t+1}, \dots$. They are connected to one another throughout time by periodic connections. In figure 17, the idea of an unrolled structure for RNN networks is depicted as a "Unfolded" form for

the situation of multiple input time steps $x_{t-1}, x_t, x_{t+1}, \dots$, multiple internal state time steps $h_{t-1}, h_t, h_{t+1}, \dots$, and multiple output time steps $y_{t-1}, y_t, y_{t+1}, \dots$.

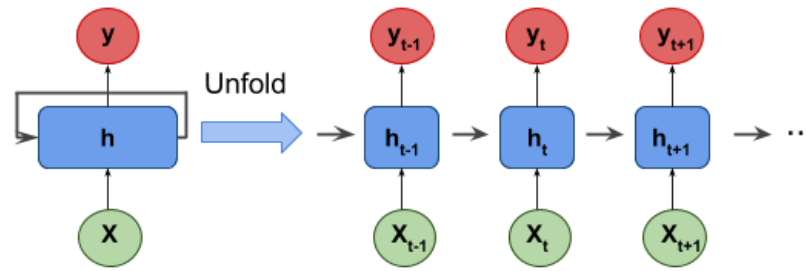


Figure 17 – Simple RNN structure and its unfolded (unrolled) form

The performance and stability of the network can be increased by initializing hidden modules with modest non-zero elements. There are disadvantages to these networks. The gradient disappearing and explosion difficulties are Simple RNN's primary drawbacks. In the process of solving the shortcomings of the standard RNN network, LSTM networks appeared. The next subsection will provide the basis of Long-term short-term memory (LSTM) networks [20, p. 6].

4.2.2 Long-term short-term memory (LSTM) for sound recognition

The Long Short-Term Memory (LSTM) architecture of the recurrent neural network (RNN) was created to address the regular RNN's vanishing and expanding gradient issues [50, p. 339-340; 51, p. 11]. In the areas of handwriting recognition, language modeling, image captioning, and classification of acoustic signals, LSTMs have demonstrated effectiveness in predicting sequence issues [87-89]. Compared to other techniques, LSTM network model training is more accurate but takes more time. Gated Recurrent Unit (GRU) networks have been designed to shorten training times while retaining a high level of training accuracy. The usage of GRU networks in classification tasks is also widespread [48, p. 2163-2-2163-10]. This thesis suggested to investigate SimpleRNN, LSTM-based RNN units, and Gated Recurrent Unit (GRU) models for UAV acoustic representations categorization challenge. Recently these models have been applied more effectively to the training of sound-based recognition systems.

A recurrent neural network structure called LSTM substitutes the normal layers of the neural system with long-term memory cell blocks to get around the issue of long-term reliance (figures 18, 19). Typical LSTM cell blocks are composed of four interlocking layers: a cell state, an input gate, an output gate, and a forget gate.

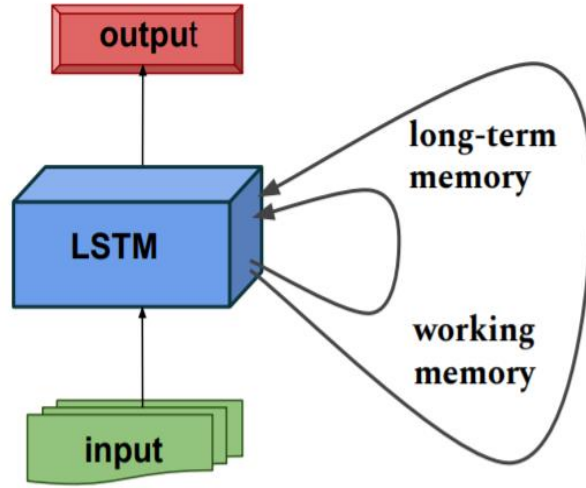


Figure 18 – LSTM architecture

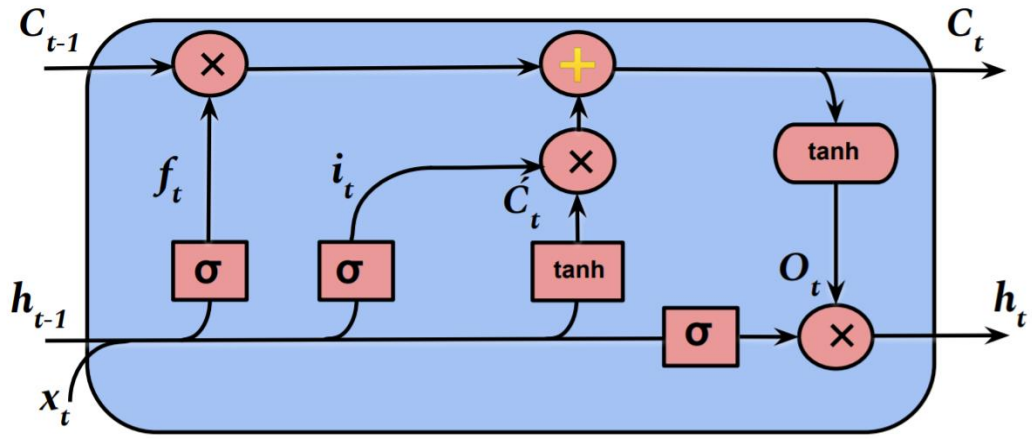


Figure 19 – Computational cell of LSTM

The output data from the previous cell h_{t-1} is mixed with the feature extraction sequence data x_t . Also, this combination of input data passes via the input gate i_t (10) and the forget gate f_t (9). Both gates have sigmoid activation functions that outputs between 0 and 1. equations (9), (10), (11), (12), (13), (14).

$$f_t = \sigma(\omega_f[h_{t-1}, x_t] + b_f) \quad (9)$$

$$i_t = \sigma(\omega_i[h_{t-1}, x_t] + b_i) \quad (10)$$

$$\hat{C}_t = \tan(\omega_c[h_{t-1}, x_t] + b_c) \quad (11)$$

$$C_t = f_t \times C_{t-1} + i_t \times \hat{C}_t \quad (12)$$

$$O_t = \sigma(\omega_o[h_{t-1}, x_t] + b_o) \quad (13)$$

$$h_t = O_t \times \tan C_t \quad (14)$$

As a result, the input gate (11) determines which input values to update, while the forget gate (9) determines what data to delete from the cell. Moreover, the tanh layer, \hat{C}_t , compresses that mixture.

Here $\omega_f, \omega_i, \omega_C$ are the weights of the corresponding gate neurons; and b_f, b_i, b_C are the offsets for the corresponding gates. LSTM cells have an inner loop (cell state) consisting of a C_t (12) variable called a constant error carousel (CEC). The old state of cell C_{t-1} is switched to set an efficient recurrent loop with the input. The compressed combination \hat{C}_t is multiplied by the \times input data of the i_t (figure 19).

A forget gate, which chooses which data should be kept or deleted from the network, is in charge of controlling this recurring loop.

Instead of multiplying, the addition approach \oplus in this case lowers the chance of the gradient disappearing. The system then uses the tanh function to push the values between $\{-1\}$ and $\{1\}$ and multiply that result by the output of the sigmoid gate to position the cell state (12).

So, this gate (13) chooses which values from the h_t cell should be output as the actual output. In general, updating the internal state is done via the input gate and the forget gate (11). Because to their many memory slots, LSTM networks have more complicated computations and greater memory requirements than simple RNNs. It varies from traditional RNNs in that it has strong advantages over gradient vanishing as well as long-term dependence.

In the course of the experiment, vanish LSTM, (figure 20) and stacked LSTM, (figure 21) models were studied, as a result, a single-layer model was developed to spend less time on calculation.

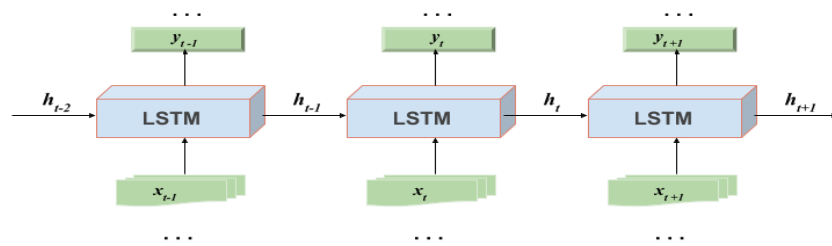


Figure 20 – Vanilla LSTM

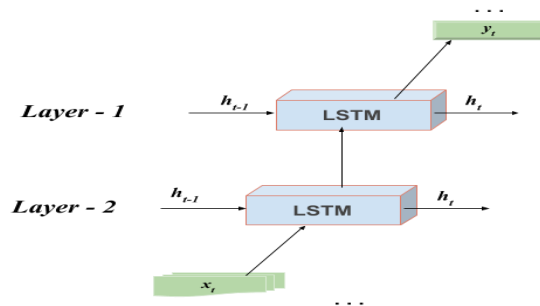


Figure 21 – Stacked LSTM

To sum up, LSTM advantages are that it overcomes disappearing and exploding gradients as well as long-term temporal dependency issues with input sequences [20, p. 6-7; 51, p. 11-12; 52, p. 2-3].

4.2.2.1 Bidirectional Long Short-Term Memory (LSTM)

Bidirectional LSTMs are a development of typical LSTMs that can enlarge model performance in sequence classification tasks. With all the time steps of the input sequence, Bidirectional LSTMs train two LSTMs instead of a single LSTM in the input. Bidirectional LSTMs solve the problem by outputting data from the input sequence in the forward and reverse directions over time steps.

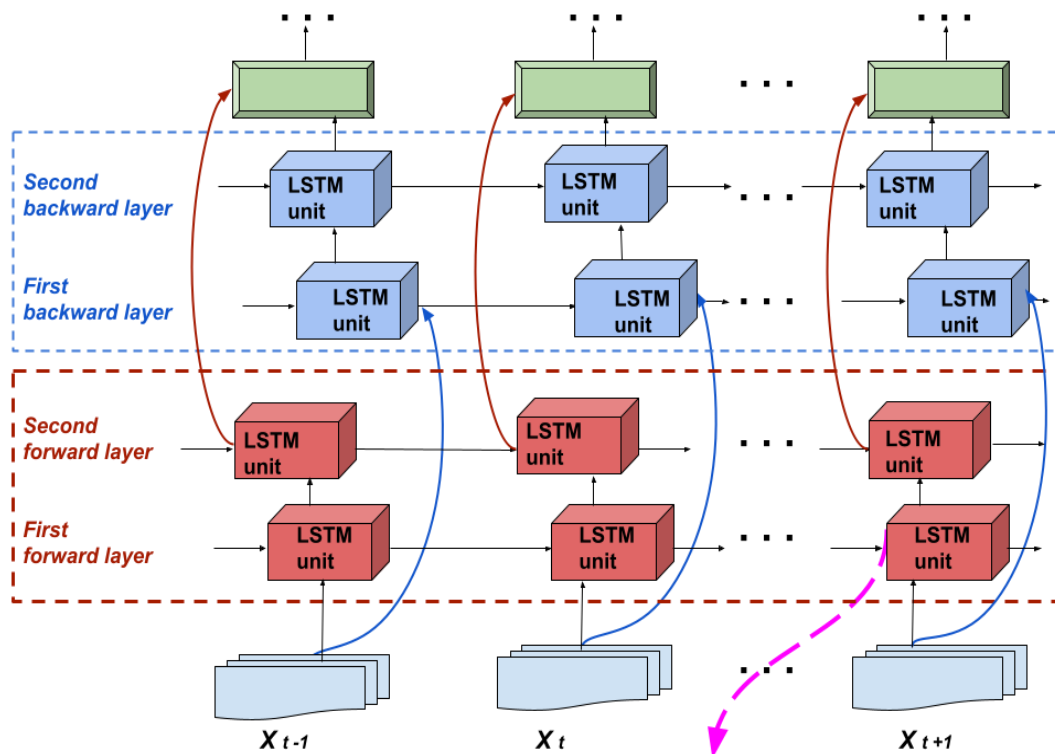


Figure 22 – Stacked Bidirectional LSTM architecture

By placing two layers side by side, delivering the input sequence exactly as it is at the first level's input, and providing a reverse copy of the input sequence at the second layer's input, this architecture effectively duplicates the first recurrence level in the network. Hence, this extra context speeds up the results [35, p. 2527-2528; 36, p. 403-405]. As a result, two different hidden layers are used by the BiLSTM network to process the x_t sequence data in both the forward and reverse directions, and their hidden layers are joined by a single output layer, as shown in figure 22. Similar to the LSTM level, the Bidirectional LSTM level's final output is a vector, $y_t = [y_{t-1}, \dots, y_{t+1}]$ the last element of which is the predicted sequence for the following time steps y_{t+1} . Due to its increased computational complexity over LSTM as a result of forward and backward learning, BiLSTM demonstrates its drawback. Their key benefit is that, compared to LSTM networks, they more accurately reflect the input sequence's past and present contexts [20, p. 7].

4.2.3 Gated Recurrent Neural Networks (GRU) for Sound Recognition

The LSTM network has been shown to be a practical solution for keeping gradients from dissipating or exploding, however because of the many memory locations in their architecture, they require more memory [49]. To address this issue, the scientists [53, p. 1724-1733] created the GRU network, which requires less learning time than the LSTM structure and still achieves great accuracy. The output gate of GRU networks is absent, in contrast to LSTM networks. The structure of GRU is seen in figure 23. Two input functions, the previous output vector h_{t-1} and the input vector x_t , are found in the structure of GRU networks at each instant of time. Moreover, the input of each gate can undergo a logical operation and a non-linear transformation before being used as the output.

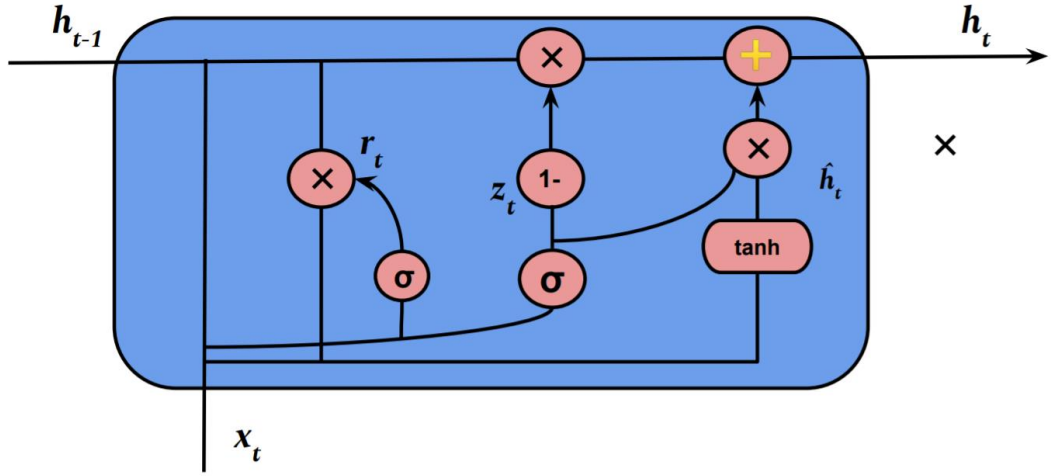


Figure 23 – Gated Recurrent Neural Network architecture

Here, the output to input ratio can be described as follows, equations (15), (16), (17), (18):

$$r_t = \sigma_g(\omega_r x_t + U_r h_{t-1} + b_r) \quad (15)$$

$$z_t = \sigma_g(\omega_z x_t + U_z h_{t-1} + b_z) \quad (16)$$

$$h_t = (1 - z_t)h_{t-1} + z_t \hat{h}_t \quad (17)$$

$$\hat{h}_t = \sigma_h(\omega_h x_t + U_h (r_t h_{t-1}) + b_h) \quad (18)$$

Where U_r , U_z , U_h and ω_r , ω_z , ω_h are weight matrices for the individual gate neurons, z_t is the update gate vector, r_t is the reset gate vectors are r_t , z_t , h_t . A hyperbolic tangent is σ_h , and σ_g is a sigmoid function [20, p. 8; 48, p. 2163-2-2163-14]).

When the reset gate is near to 0, as in this configuration (figure 23), the hidden state ignores the prior hidden state and only resets with the current input. This enables

the hidden state to delete any data that will no longer be relevant in order to provide a view that is more condensed. The update gate also regulates the amount of data that is moved from the prior hidden state to the present hidden state. This enables the RNN to store long-term information and functions similarly to LSTM memory cells. Each hidden module will develop the ability to recognize dependencies at various time scales because each one has a unique reset and update gate. A reset gate will be frequently active in modules that are taught to capture short-term dependencies, but an updated gate will be more frequently active in modules that gather long-term dependencies [53, p. 1725-1730; 90-92].

The drawback of GRU is that it has a higher computational cost and memory need than Simple RNN because to the numerous hidden state vectors. GRU networks offer a wider range of practical uses thanks to benefits like the capacity to represent long-term dependent sequences, resilience to gradient reduction, and reduced memory requirements. This thesis aimed to conduct practical research for the UAV sound recognition task, taking into consideration all the characteristics of computing modules RNN networks stated above in theory [20, p. 6-8].

5 REAL-TIME UAV ACOUSTIC DATA RECOGNITION AND CLASSIFICATION SYSTEM

5.1 The proposed real-time Drone Sound recognition system

The study of this dissertation work is aimed at developing a drone detection system with the recognition of their acoustic data. The main objective is the creation of a system that can recognize UAV sounds in real-time. This is due to the fact that when suspicious UAVs are employed in crowded areas, identifying UAV sounds aids in establishing security as one of the UAV detection methods. The key issue is real-time detection of suspicious UAVs. One of the initial steps is fast real-time analysis of their acoustic data. The literature review section demonstrates that while drone sound processing and recognition has been researched generally, but no specific work has been found to adapt it for real-time performance based on SPU or GPU base. In order to recognize UAV sounds in real time, it must be possible to quickly process their acoustic data. In this regard, our study considered it appropriate to investigate this research question.

In most sound recognition studies, the signal pre-processing step is often performed in advance during the data preparation step. That is, these pre-prepared data are stored in special folders, which occupied additional space and time, then it could be obtained from these folders when it is fed into the recognition algorithm. So, in the preparation of acoustic data of the UAV at this stage consisted of two steps: first, the UAV was recorded in different states, and then processed in advance. And according to the proposed study of this dissertation, the fast sound signal processing layer is located before the neural layers as the Keras layer. It is based on the KAPRE method [43]. This section creates a proposed recognition system through the next three subsections: “Adaptation of UAV Sound Recordings for Real Time System”, “Processing of UAV acoustic signals using the KAPRE method: Melspectrogram” and “Real-time and RNN network-based UAV sound recognition architecture”.

5.1.1 Adaptation of UAV Sound Recordings for Real Time System

In general, the study of the sound recognition of drones began with the recording of their sounds. This is due to the fact that the sounds of the drones were needed as initial data to start the study. This procedure was therefore carried out and addressed prior to the selection of methods and their theoretical explanation in section 2. The main focus of this dissertation is the development of a recognition system that can detect in real-time. "Real-time" comprehension in this study is presented as a system adapted to recognize an audio file of 1 second duration. To do this, it must first satisfy the requirements for processing audio data using deep learning. In other words, audio data for deep learning model needs to be divided into groups for "training" to train the model and "validation" to test the model's reliability. Firstly, all the recorded sounds of drones were divided into 3 main classes in accordance with their content value in Section 2. It was a class of “Loaded UAV” with a special payload imitating suspicious drones, and the class “Unloaded UAV” and the “Background noise” class. The collected sounds for these 3 different classes

were preserved with their initial length of 3 different folders. However, the duration of the audio recording must be adjusted such that recognition feedback can be received every second. Therefore, the length of previously recorded and collected audio data is requested to be 1 second each. Because of this task, a “special filtering block” has been created that re-adapts recorded UAV audio recordings of various states and lengths to the given conditions into appropriate folders (figure 24).

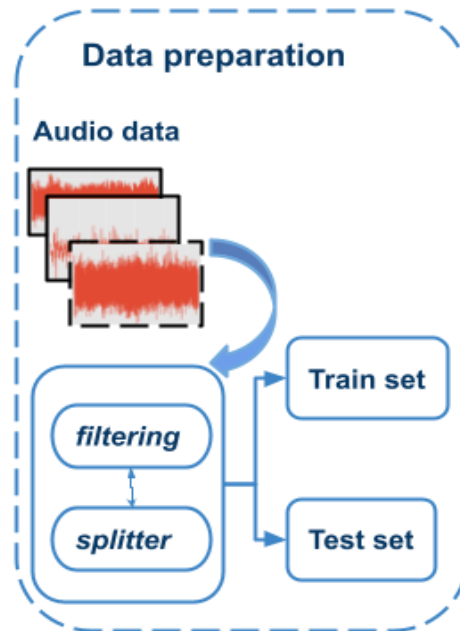


Figure 24 – UAV sound recording adaption algorithm for real-time systems

This “filtering block” retrieves all previously recorded UAV audio files of 1-second length. It is important to emphasize that the study was carried out using supervised learning. The filtering unit receives audio data of different lengths. There are 2 functions here. One of them is envelope function (figure 25). And threshold value taken as "0". Because the envelope of an oscillating signal is a smooth curve defining its extremes, the envelope function was utilized (figure 26). Since the sounds of the drone are superimposed on background sounds and there are sounds from various motorized objects, the “0” threshold was effective.

```
def envelope(y, rate, threshold):
    mask = []
    y = pd.Series(y).apply(np.abs)
    y_mean = y.rolling(window=int(rate/20),
                      min_periods=1,
                      center=True).max()
    for mean in y_mean:
        if mean > threshold:
            mask.append(True)
        else:
            mask.append(False)
    return mask, y_mean
```

Figure 25 – Implementation of the envelope function in “Filtering Block”

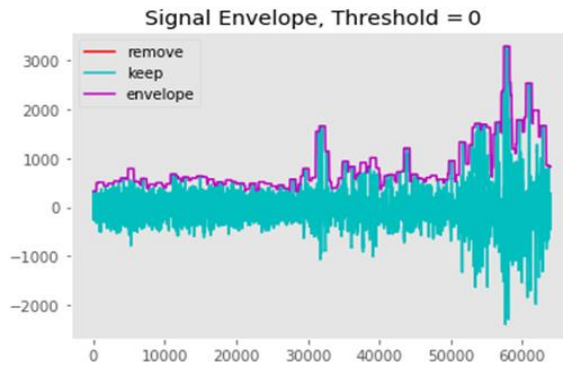


Figure 26 – Signal Envelope method with the threshold “0”

All initial files of their different lengths, figure in (Appendix E), were cut as 1-second audio files and stored in folders classified according to the initial classes such as “Loaded UAV”, “Unloaded UAV”, and “Background noise” (figure 27).

```

parser.add_argument('--delta_time', '-dt', type=float, default=1.0,
                    help='time in seconds to sample audio')
a

def save_sample(sample, rate, target_dir, fn, ix):
    fn = fn.split('.wav')[0]
    dst_path = os.path.join(target_dir.split('.')[0], fn+'_{}.wav'.format(str(ix)))
    if os.path.exists(dst_path):
        return
    wavfile.write(dst_path, rate, sample)
b

```

a – Splitting long files into seconds files; b – Saving received files with classes

Figure 27 – Audio filter preserving audio files under one second in length

The acoustic data of the UAV were adapted before studying the stage of recognition of UAV sounds based on the analysis of frequency ranges. Acoustic data was studied in the time domain first. And our background noise class consists of the sounds of many motorized objects. The sounds of these objects were collected to prevent false recognition due to the possibility of confusion during recognition. Therefore, due to the large number of types of background noise objects, the background noise class was temporarily expanded in this adaptation step, Figures in (Appendix D) and figures 28, 29. The frequency range of the extended classes was then studied to preliminarily determine the range of object spectra up to the KAPRE layers in the model. Therefore, it is necessary to analyze the sounds of various objects in the frequency domain based on their natural appearance (figures 28, 29).

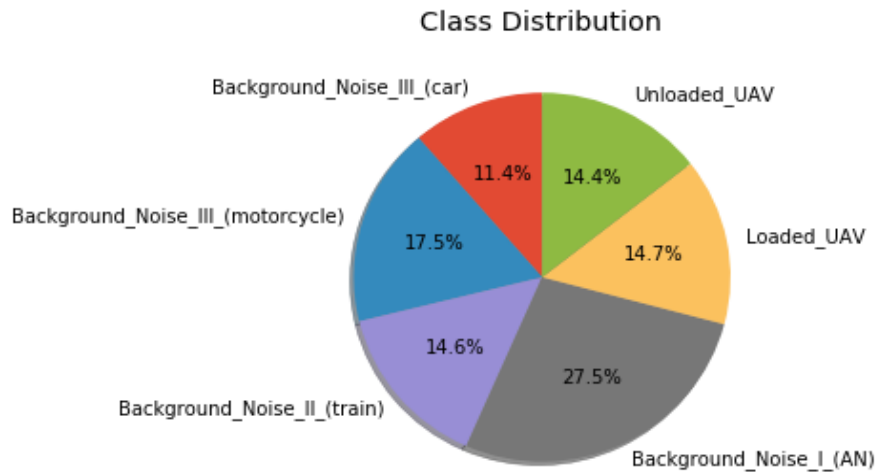


Figure 28 – Temporary extension of background noises

The spectra for each class in the frequency range for our signals were obtained using the Fast Fourier Transform (FFT) to perform this fundamental analytical work. They were separated in time and frequency domain only temporarily during the adaptation analysis stage. And during the application in the neural model, these all extended classes of background noise were processed together as background noise.

This class extension analysis method helped to determine the frequency ranges of the desired objects at the level of 16000 Hz, since the informative parts were visible only in this region (figure 29). That is, the informative component of the sounds of the UAV and the background noise we need is reflected only up to 16000 Hz, which can be seen in the frequency domain (figure 29).

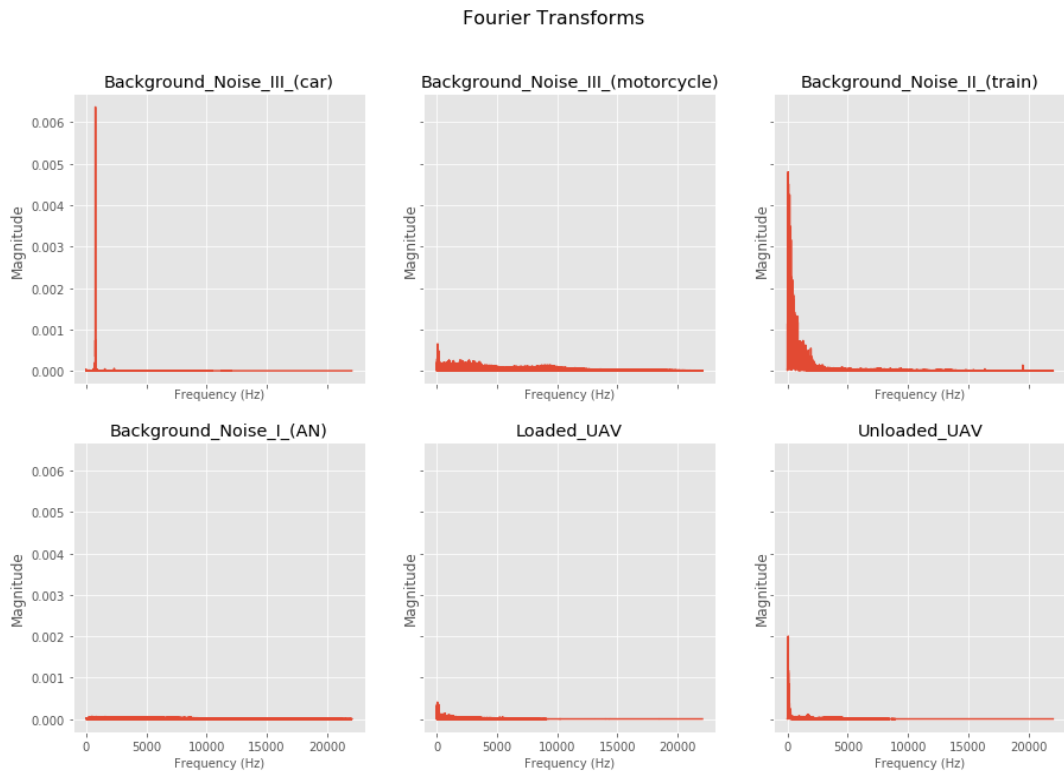


Figure 29 – UAV signal analysis in the frequency domain using extended six classes

The UAV dataset had adapted to be down sampled based on research of objects' frequency range. This specially created filter unit of the Downsampling was constructed to perform these tasks (figure 30).

```
def downsample_mono(path, sr):
    rate, wav = wavfile.read(path)
    wav = resample(wav.astype(np.float32), rate, sr)
    wav = wav.astype(np.int16)
    try:
        tmp = wav.shape[1]
        wav = wav[:,0]+wav[:,1] / 2
    except:
        pass
    return sr, wav
```

Figure 30 – Function unit for "Downsampling"

The audio data through this block gave a database of audio files with 1 second duration and a frequency set to 16000 Hz (figure 31). In addition, cutting off the audio spectrum above the 16,000 Hz region saved over-computing time.

```
if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Audio Classification Training')
    parser.add_argument('--model_type', type=str, default='conv2d',
                        help='model to run. i.e. conv1d, conv2d, SimpleRNN, GRU')
    parser.add_argument('--src_root', type=str, default='clean',
                        help='directory of audio files in total duration')
    parser.add_argument('--batch_size', type=int, default=16,
                        help='batch size')
    parser.add_argument('--delta_time', '-dt', type=float, default=1.0,
                        help='time in seconds to sample audio')
    parser.add_argument('--sample_rate', '-sr', type=int, default=16000,
                        help='sample rate of clean audio')
    args, _ = parser.parse_known_args()
    train(args)
```

Figure 31 – Filtering block with Downsampling

The characteristics of the audio signal of the UAV at this stage of the temporal expansion of classes are not processed, but only adapted. The spectra of the signals that have successfully adapted to the above procedure using the filter bank are shown in Figures (Appendix F, G). Feature extraction from audio signals has been incorporated into the deep learning model itself, which will be discussed in the next section. The next subsection discusses building the first layer of a basic RNN recognition model, i.e. the signal processing layer, using the Keras libraries and the KAPRE method.

5.1.2 Processing of UAV acoustic signals using the KAPRE method: Melspectrogram

The deep neural network models that will be presented in this paper in the next subsection use the image-based classification method. And it is able to distinguish between different types of object images according to their feature vectors taken from the audio data. Therefore, it is necessary to extract feature vectors from UAV sounds.

By examining the frequency spectrum of drone sounds extensively, these feature vectors can be produced. In general, the processing of drone sound data obtained during research in the frequency range is called "feature extraction". Efficient frequency extraction for a real-time UAV sound recognition system was found in the course of empirical studies that were published earlier in publications [15, p. 457-458; 16, p. 473-474; 20, p. 26-24-26-25]. Efficient frequency extraction was the layers of Melspectrogram [20, p 26-25]. Table 5 displays the hyperparameter ranges and chosen values for the Melspectrogram feature layer. The python programming environment was used to perform fast calculations to obtain this Melspectrogram layer.

Table 5 – Hyperparameters of Melspectrogram layer

Keras layers	Hyperparameters	Best fit	Range
Melspectrogram	Sampling rate	16000 Hz	600-44100 Hz
	Window length	512	512, 1024
	Hop length	160	160, 256
	Number of Mels	128	40-128
	[Frequency, Time]	128*100	

Thus, it is suggested that the vectors of the Mel scale be extracted from the UAV sound data while keeping the time and frequency information parameters, which are called STFTs. In many investigations, the libraries Librosa and Essentia are primarily used to implement temporal and frequency characteristics based on conventional approaches. This study implements the KAPRE method built as Keras layers in Python. The adjustment of acoustic sound processing parameters is the main benefit of the Kapre approach. And the presentation of the hyperparameters of this layer from a programmatic point of view was given in (figure 32a).

```
input_shape = (int(SR*DT), 1)
i = get_melspectrogram_layer(input_shape=input_shape,
                             n_mels=128,
                             pad_end=True,
                             n_fft=512,
                             win_length=400,
                             hop_length=160,
                             sample_rate=SR,
                             return_decibel=True,
                             input_data_format='channels_last',
                             output_data_format='channels_last',
                             name='2d_convolution')
```

a

Layer (type)	Output Shape	Param #	Connected to
stft_3_input (InputLayer)	[(None, 16000, 1)]	0	
stft_3 (STFT)	(None, 100, 257, 1)	0	stft_3_input[0][0]
magnitude_3 (Magnitude)	(None, 100, 257, 1)	0	stft_3[0][0]
apply_filterbank_3 (ApplyFilter)	(None, 100, 128, 1)	0	magnitude_3[0][0]
magnitude_to_decibel_3 (Magnitu	(None, 100, 128, 1)	0	apply_filterbank_3[0][0]

b

a – Programmatically feeding the hyperparameters of the Melspectrogram layer; b – Layers of processed signals based on Melspectrogram Layer during training

Figure 32 – Implementation of the Melspectrogram Keras layer

And when the hyperparameters of this Melspectrogram layer are implemented in accordance with the code line in figure 32a, the mathematical calculations which discussed in the third section above will be performed based on fast calculations. This can be seen from (figure 32b).

The features of the proposed Melspectrogram layer can also be seen visually in figure 33, where they are represented as a picture.

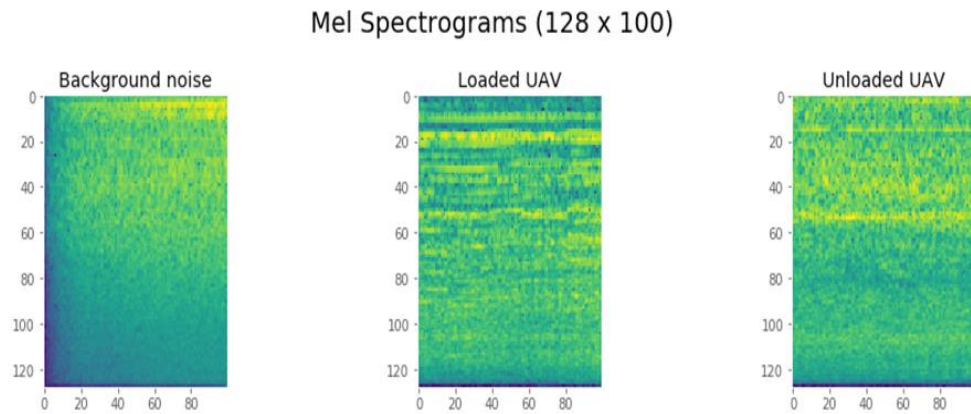


Figure 33 – Melspectrogram images with 4 class database case

The other representations of feature extraction methods such as MFCC, Filter bank, their normalization, and data augmentation can also be carried out in real-time on a GPU or CPU using this KAPRE method. In fact, this method allows finding optimal time-frequency representations and their characteristics for use in audio pre-processing. This can save a lot of memory since each configuration often uses the same amount of memory as the decoded audio samples [20, p. 7-8]. A wide variety of methods of the feature Extraction were obtained and used during empirical research. And their visual appearances are presented in the figures in the appendices section. The sequence of experiments demonstrated the effectiveness of the Melspectrogram layer. Therefore, the Melspectrogram layer was chosen for the drone database.

To summarize this subsection, the Melspectrogram is a KAPRE layer that has been expanded on the spectrogram by multiplying the Mel scale transformation matrix from the linear frequencies [43]. The proposed approach explored a large range for the Melspectrogram layer, as can be seen from (table 5). As a result of experimental attempts, 100-time vectors and 128 frequency features were obtained, (figure 33). And the next subsection will consider the entire structure of the proposed algorithm.

5.1.3 Real-time and RNN network-based UAV sound recognition architecture

This dissertation is primarily aimed at performing real-time UAV sound classification. The study of CNN and RNN networks, which are widely known for general purpose object recognition systems and have a high recognition ability, is considered. According to literature reviews, CNNs have been the preferred models for image processing and recognition. Much of the work on UAV sound recognition using CNN models has been achieved by deepening the CNN layers. Again, a

number of references show that most sound recognition systems are the preferred models for processing and recognizing audio signals. On this basis, extensive experimental studies have been carried out on the recognition ability of recurrent neural networks compared with CNN networks. All common types of recurrent networks have been considered. In particular, SimpleRNN, LSTM, Bidirectional LSTM and GRU networks. The significant success of recurrent neural networks in the analysis of sound (audio) signals and speech has become a motivating factor for their more extensive practical research on sounds than CNNs. Because the initial data in this thesis are sound data, not images.

Previous studies [1, p. 863-865; 2, p. 244-245] on UAV sound recognition have shown that signals of 20 seconds or less were processed and recognized. That is, using the research model of work [1, p. 865-866] would require 20 seconds to process sound for the security of the protected environment. In general, this work is appreciated because it has made a great contribution to the solution of the scientific question of the recognition of drone audio signals and is able to experimentally prove the possibility of UAV audio data using one model of the Phantom series. In general, many dangerous situations could develop in 20 seconds. That is why it is necessary to create a real-time system that could instantly distinguish between UAV states, in particular, loaded and unloaded UAVs or background noise. These aspects were taken into account when carrying out experimental work on the problem of classifying three types of UAV sounds using four different types of computational neural cells from recurrent neural networks, including SimpleRNN, LSTM, Bidirectional LSTM and GRU. The classes "Unloaded UAV", "Loaded UAV" and "Background noise" were taken as the main base classes. This is because in many situations, identifying a particular type of drone does not result in a pressing need. However, earlier works [6, p. 2-3; 7, p. 1-3] considered the potential of this problem. Also, figuring out the drone's load is an extremely important system for scenarios that seem suspicious. In particular, it can be applied as a solution or preventive action for life-threatening problems such as the transit of life-threatening products, the possibility of weight being dropped on people even though it is harmless, and for military purposes. Also, the detection of a suspicious UAV entering a strategically protected area can be resolved with the ability to recognize UAV sounds. Furthermore, figuring out the UAV's load is challenging because it sounds similar to the UAV itself. The flight sound with an extra weight might have different time and frequency matrices, which might facilitate recognition. Therefore, in order to find a solution to this scientific question, the architecture of a deep learning based neural network was studied from an experimental point of view. Based on extensive research, a deep neural network architecture has been developed for recognizing UAV acoustic data. And in this proposed system, the UAV sound processing step is added as the first layers of the recognition system architecture. The structure was given in figure 34 [20, p. 11].

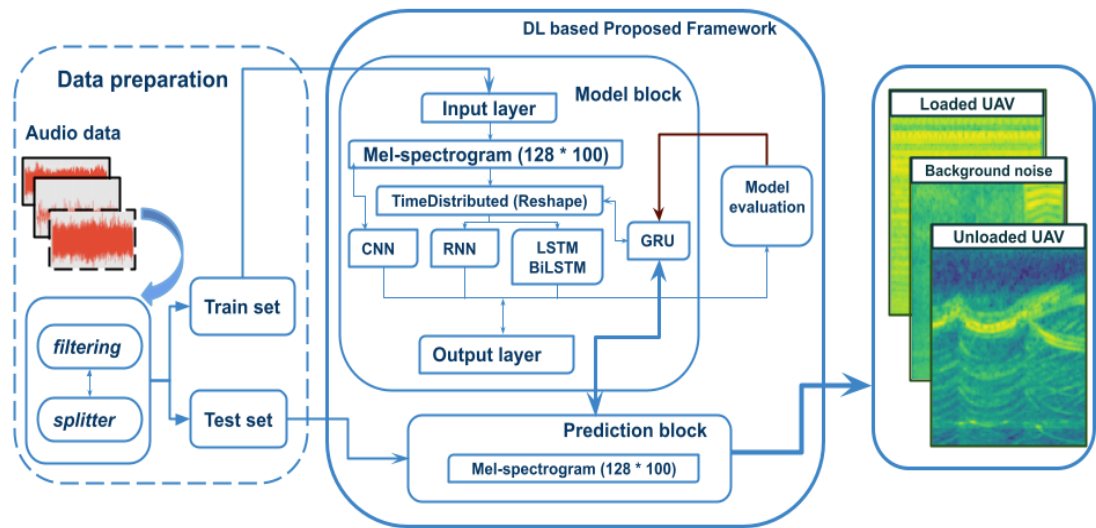


Figure 34 – The proposed RNN based Framework for Real-time UAV sound recognition

It can be seen from figure 34 that in the proposed structure, the block on the left is a 1 second file type adaptor, which filters the audio data as explained in the previous subsections. This device can be thought of as a drone sound production stage. And the main big block in the middle of figure 34 is a deep learning structure with modified Melspectrogram that allows us to recognize drone sounds.

The input layer of this main block is the Melspectrogram layer, which processes drone sounds based on the On-CPU. The Melspectrogram is processed with the help of STFT calculation and FFT calculation in real time according to the respective steps as explained in the theoretical framework. Therefore, this layer consists of several layers during training.

And from one layer during the code line. This layer calculates features of UAV acoustic data in the dimension of 128 features vectors of frequency by 100 features vectors of the time. And the KAPRE method libraries in Keras, which allow processing this layer, are pre-installed and their libraries are called according to the programming requirements. The tuned hyperparameters of the proposed UAV acoustic data recognition architecture was given in table 6.

Table 6 – The hyperparameters of the proposed architecture

Keras layers	Hyperparameters	Best fit	Range
1	2	3	4
Melspectrogram	Sampling rate Window length Hop length Number of Mels [Frequency, Time]	16000 Hz 512 160 128 128*100	600-44100 Hz 512, 1024 160, 256 40-128
LayerNormalization	Batch Normalization	-	-
Reshape	TimeDistributed (Reshape)	-	-
Dense	TimeDistributed (Dense), tanh	-	-

Table 6 continuation

1	2	3	4
RNN cells (Fmaps)	SimpleRNN, LSTM, BiLSTM, GRU	<i>GRU (64)</i>	32-64
Concatenate			
Dense	Dense, relu	<i>(64)</i>	33-128
MaxPooling	MaxPooling1D	-	-
Dense	Dense, relu	<i>32</i>	32-128
Flatten			
Dropout	Dropout	<i>0.25</i>	0.2-0.3
Dense	Dense, relu activity regularizer activity regularizer	<i>32</i> <i>0.01</i> <i>0.00001 for</i> <i>GRU (64)</i>	8-32 0.01-0.00001 0.01-0.00001
Dense	Dense Activation in classification Optimization solver # epochs	<i>(# classes) 3</i> <i>softmax</i> <i>adam</i> <i>25</i>	(3,4) sgdm, adam 25-150

Melspectrogram hyperparameters were studied experimentally in order to determine the effective length of the feature extraction of this layer in different ranges. 100 and 128 are the most effective lengths. And the studied range of the lengths of the vectors of this layer were discussed in table 5, 6. And it was modified by splitting frames according to table 5 from 1 second audio. This is because deepening other layers in the recognition architecture created with deep learning did not yield a very high recognition capability. In this regard, 2 factors should be mentioned. First, the UAV acoustic data was investigated with a small database at the beginning, figures (Appendix C). Secondly, in previous publications [15, p. 455; 16, p. 472], the MFCC signal processing method was used with short feature vectors for recognition. And then the Melspectrogram layer was studied extensively. On the basis of this layer, a sufficiently high recognition rate was obtained. So, the modified feature vector was selected according to Table 5 and 6. The UAV acoustic database was expanded from the initial. The study was carried out again on this database using modified Melspectrogram, and published in work [20, p. 18]. To adapt acoustic data feature vectors derived from Melspectrogram layers to feed into RNN layers, normalization and layer reshaping layers were provided (table 6). These Melspectrograms use a normalization layer after themselves in the model that normalizes the 2D input data by time, frequency, batch and channel. Further, the received vectors are sent through the TimeDistributed (Dense) layer with the activation function tanh and fed into the RNN cells. The recurrent cells SimpleRNN, LSTM, BiLSTM, GRU, described in the theoretical section, were used as RNN layers. As a result, considering each type of RNN models separately, 4 RNN models were studied. A concatenation layer was added after the RNN cells, and dense layers were connected depending on the hyperparameters as in Table 6. The number of cells of recurrent networks was initially taken as 32. To simplify the study of the design, the MaxPooling1D layer was added. Then a dense layer of 32 relu cells was

developed. Multidimensional output is also linearized and transferred to a dense layer using a Flatten layer. For the classification task, the output of the Flatten layer is passed to the next layers. When testing a 32-cell RNN model, a Dropout layer with a coefficient of “0.2” was added as a next layer to prevent model overfitting. Before the final dense layer, a 32-cell dense layer was added along with an activity regularizer and a “relu” function. It is important to note that the activity regularizer feature significantly affected the to the accuracy plots during model training for certain UAV sounds. Thus, the range of this function was from 0.01 to 0.00001. And the coefficient of Dropout layer has been adjusted according to the size of the RNN cells. In the case of the GRU model, a change in the dropout coefficient from 0.2 to 0.3 was taken into account, since the model could be retrained with an increase in the number of cells to 64. As a result, a factor of 0.25 was optimal for the Dropout layer in GRU model case.

The "categorical cross-entropy" loss function is tuned for the multiple classification problem in the model implementation. The classification problem and weights are optimized using the "Adam" gradient descent implementation. To assess the model's ability to learn and generalize across all architectures, "accuracy" is calculated during model training and validation.

The proposed deep learning-based recognition architecture was designed with hyperparameter tuning set according to table 6. A total of 5 models were considered, including 4 RNN models and 1 CNN model. The results obtained and visible layers during training will be discussed in more detail in the next section. The proposed deep learning framework, trained according to the hyperparameters in table 6, consisted of only one RNN layer. Compared to previous studies [1, p. 862-865; 15, p. 457; 16, p. 473-474], it differs in that it has a simpler structure. This, in turn, requires less computation for calculations. The number of epochs indicating the number of training sessions is also small. However, many experimental studies have been carried out empirically to determine the hyperparameters of such a simple and fast computational structure of the CNN as well. This set of experiments is shown in Table 6 as the hyperparameter search area. And information about the recognition process trained on the basis of this modified model and its verification is widely discussed. The results of these studies are summarized in the Learning and Assessment Metrics subsection. However, the CNN shows good recognition capability only by increasing the CNN layers. This model deepened with several CNNs itself had a lower recognition rate than a single-layer RNN network. For this reason, in this dissertation, priority is given to the study of the ability to recognize types of RNNs. And in order to justify the comparison of the RNN network with the results of the CNN network, the subsection will be discussed based on the experimental results.

5.2 Results and discussion of the Proposed System

In this subsection, four recurrent neural network models such as SimpleRNN, LSTM, BiLSTM, and GRU were trained using the proposed neural architecture from (table 6). And the CNN model structure shown in table 5 of [1, p. 243] was trained

using our set data. During the experimental work, the tests were carried out on the Python program (Appendix H) and on the Intel(R) Core (TM) i5-8265U processor with a clock frequency of 1.60 GHz. The distribution of UAV acoustic data is divided according to 70 by 30, 70% of the total number of received sound files were given for training, and the remaining 30% were stored for validation, which were not seen by the models. In addition to these 30% validation files, 100 files of 1 second duration for each class were separately saved, for a total of 300 files for all three classes, to see how well and consistently the models can distinguish between each class. As a result of training, the recognition accuracy of models developed for real-time systems was thoroughly tested on 30 percent of the training data using model recognition accuracy plots. Initially, all models were trained with 30 epochs when they had 32 RNN cells on models. The "good fit" model curve area was selected with 25 epochs after the training and having the accuracy plots. Each model went through a new training run of 25 epochs before being saved as a Pickle file with a ".h5" extension. And the classification efficiency of 300 independently stored files was tested using the confusion matrix, F1, recall, and accuracy. The training times for these models are given in figure 35 pictures in seconds.

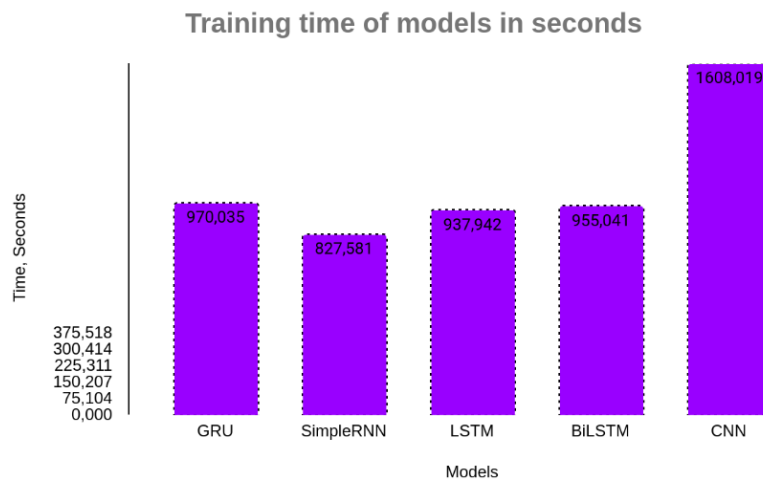


Figure 35 – Training time with proposed models

To evaluate the practical applicability of RNN cell type recognition capabilities, all layers of the architecture of RNN models remained unchanged after training. Two experimental tests were run on these trained RNN models. When training each model, the following architectures were obtained. Due to the fact that the priority of the dissertation research is given to the architecture consisting of RNN networks, the figure below shows the architecture of RNN models obtained after training, figures 36, 37, 38, 39. At the initial stage, the graphs of the accuracy of the model were studied. This was necessary to evaluate the reliability of the trained models on the given dataset. The second step included a detailed prediction for 300 "one-second audio files" that had previously been stored separately. The confusion matrix was obtained to accurately assess the ability to recognize individual classes.


```

Model: "SimpleRNN"
-----
Layer (type)                Output Shape                Param #   Connected to
-----
stft_3_input (InputLayer)    [(None, 16000, 1)]         0         stft_3_input[0][0]
stft_3 (STFT)                (None, 100, 257, 1)       0         stft_3_input[0][0]
magnitude_3 (Magnitude)      (None, 100, 257, 1)       0         stft_3[0][0]
apply_filterbank_3 (ApplyFilter) (None, 100, 128, 1)       0         magnitude_3[0][0]
magnitude_to_decibel_3 (Magnitu) (None, 100, 128, 1)       0         apply_filterbank_3[0][0]
batch_norm (LayerNormalization) (None, 100, 128, 1)       256        magnitude_to_decibel_3[0][0]
reshape (TimeDistributed)    (None, 100, 128)          0         batch_norm[0][0]
td_dense_tanh (TimeDistributed) (None, 100, 64)          8256        reshape[0][0]
SimpleRNN (SimpleRNN)        (None, 100, 32)           3104        td_dense_tanh[0][0]
skip_connection (Concatenate) (None, 100, 96)          0         td_dense_tanh[0][0]
                                SimpleRNN[0][0]
dense_1_relu (Dense)         (None, 100, 64)           6208        skip_connection[0][0]
max_pool_1d (MaxPooling1D)   (None, 50, 64)            0         dense_1_relu[0][0]
dense_2_relu (Dense)         (None, 50, 32)            2080        max_pool_1d[0][0]
flatten (Flatten)            (None, 1600)              0         dense_2_relu[0][0]
dropout (Dropout)            (None, 1600)              0         flatten[0][0]
dense_3_relu (Dense)         (None, 32)                51232       dropout[0][0]
softmax (Dense)              (None, 3)                 99         dense_3_relu[0][0]
-----
Total params: 71,235
Trainable params: 71,235
Non-trainable params: 0
-----
IPython Console History

```

Figure 36 – The architecture obtained during the compilation of a simple RNN network

```

Model: "LSTM"
-----
Layer (type)                Output Shape                Param #   Connected to
-----
stft_2_input (InputLayer)    [(None, 16000, 1)]         0         stft_2_input[0][0]
stft_2 (STFT)                (None, 100, 257, 1)       0         stft_2_input[0][0]
magnitude_2 (Magnitude)      (None, 100, 257, 1)       0         stft_2[0][0]
apply_filterbank_2 (ApplyFilter) (None, 100, 128, 1)       0         magnitude_2[0][0]
magnitude_to_decibel_2 (Magnitu) (None, 100, 128, 1)       0         apply_filterbank_2[0][0]
batch_norm (LayerNormalization) (None, 100, 128, 1)       256        magnitude_to_decibel_2[0][0]
reshape (TimeDistributed)    (None, 100, 128)          0         batch_norm[0][0]
td_dense_tanh (TimeDistributed) (None, 100, 64)          8256        reshape[0][0]
LSTM (LSTM)                  (None, 100, 32)           12416       td_dense_tanh[0][0]
skip_connection (Concatenate) (None, 100, 96)          0         td_dense_tanh[0][0]
                                LSTM[0][0]
dense_1_relu (Dense)         (None, 100, 64)           6208        skip_connection[0][0]
max_pool_1d (MaxPooling1D)   (None, 50, 64)            0         dense_1_relu[0][0]
dense_2_relu (Dense)         (None, 50, 32)            2080        max_pool_1d[0][0]
flatten (Flatten)            (None, 1600)              0         dense_2_relu[0][0]
dropout (Dropout)            (None, 1600)              0         flatten[0][0]
dense_3_relu (Dense)         (None, 32)                51232       dropout[0][0]
softmax (Dense)              (None, 3)                 99         dense_3_relu[0][0]
-----
Total params: 80,547
Trainable params: 80,547
Non-trainable params: 0
-----
IPython Console History

```

Figure 37 – The architecture obtained during the compilation of a simple RNN network


```

Model: "BiLSTM"
-----
Layer (type)                Output Shape                Param #   Connected to
-----
stft_2_input (InputLayer)    [(None, 16000, 1)]         0         (stft_2_input[0])[0]
stft_2 (STFT)                (None, 100, 257, 1)       0         stft_2_input[0][0]
magnitude_2 (Magnitude)      (None, 100, 257, 1)       0         stft_2[0][0]
apply_filterbank_2 (ApplyFilter) (None, 100, 128, 1)       0         magnitude_2[0][0]
magnitude_to_decibel_2 (Magnitu) (None, 100, 128, 1)       0         apply_filterbank_2[0][0]
batch_norm (LayerNormalization) (None, 100, 128, 1)       256        magnitude_to_decibel_2[0][0]
reshape (TimeDistributed)    (None, 100, 128)          0         batch_norm[0][0]
td_dense_tanh (TimeDistributed) (None, 100, 64)          8256        reshape[0][0]
bidirectional_lstm (Bidirection) (None, 100, 64)          24832       td_dense_tanh[0][0]
skip_connection (Concatenate) (None, 100, 128)         0         td_dense_tanh[0][0]
bidirectional_lstm[0][0]
dense_1_relu (Dense)         (None, 100, 64)          8256        skip_connection[0][0]
max_pool_id (MaxPooling1D)   (None, 50, 64)           0         dense_1_relu[0][0]
dense_2_relu (Dense)         (None, 50, 32)           2080        max_pool_id[0][0]
Flatten (Flatten)           (None, 1600)              0         dense_2_relu[0][0]
dropout (Dropout)           (None, 1600)              0         flatten[0][0]
dense_3_relu (Dense)         (None, 32)                51232       dropout[0][0]
softmax (Dense)              (None, 3)                 99         dense_3_relu[0][0]
-----
Total params: 95,011
Trainable params: 95,011
Non-trainable params: 0
-----
IPython Console History

```

Figure 38 – The architecture obtained during the compilation of a simple RNN network

```

Model: "GRU"
-----
Layer (type)                Output Shape                Param #   Connected to
-----
stft_2_input (InputLayer)    [(None, 16000, 1)]         0         (stft_2_input[0])[0]
stft_2 (STFT)                (None, 100, 257, 1)       0         stft_2_input[0][0]
magnitude_2 (Magnitude)      (None, 100, 257, 1)       0         stft_2[0][0]
apply_filterbank_2 (ApplyFilter) (None, 100, 128, 1)       0         magnitude_2[0][0]
magnitude_to_decibel_2 (Magnitu) (None, 100, 128, 1)       0         apply_filterbank_2[0][0]
batch_norm (LayerNormalization) (None, 100, 128, 1)       256        magnitude_to_decibel_2[0][0]
reshape (TimeDistributed)    (None, 100, 128)          0         batch_norm[0][0]
td_dense_tanh (TimeDistributed) (None, 100, 64)          8256        reshape[0][0]
GRU (GRU)                    (None, 100, 64)          24960        td_dense_tanh[0][0]
skip_connection (Concatenate) (None, 100, 128)         0         td_dense_tanh[0][0]
GRU[0][0]
dense_1_relu (Dense)         (None, 100, 64)          8256        skip_connection[0][0]
max_pool_id (MaxPooling1D)   (None, 50, 64)           0         dense_1_relu[0][0]
dense_2_relu (Dense)         (None, 50, 32)           2080        max_pool_id[0][0]
Flatten (Flatten)           (None, 1600)              0         dense_2_relu[0][0]
dropout (Dropout)           (None, 1600)              0         flatten[0][0]
dense_3_relu (Dense)         (None, 32)                51232       dropout[0][0]
softmax (Dense)              (None, 3)                 99         dense_3_relu[0][0]
-----
Total params: 95,139
Trainable params: 95,139
Non-trainable params: 0
-----
IPython Console History

```

Figure 39 – The architecture obtained during the compilation of a simple RNN network

Table 7 below shows the average value of the recognition results of the first 32 cell RNN networks. Also, the plot of the recognition accuracy obtained with the performance of this training in each epoch is given in figure 36. In this table 7, the results of the recognition accuracy obtained by the CNN model are also given

Table 7 – Comparison of Model Accuracy of the SimpleRNN, LSTM, BiLSTM, GRU, and CNN models on 128-100 dimensional Melspectrograms

Trained models	Accuracies in %
SimpleRNN	98
LSTM	97
Bidirectional LSTM (BiLSTM)	97
GRU with 32 cells	98
CNN structure as in [1]	94
GRU with 64 cells	98

Accuracy plots of the models created by their values from 25 epochs. Here, the solid lines represent the training line, and the dotted line represents the test line, (figure 40).

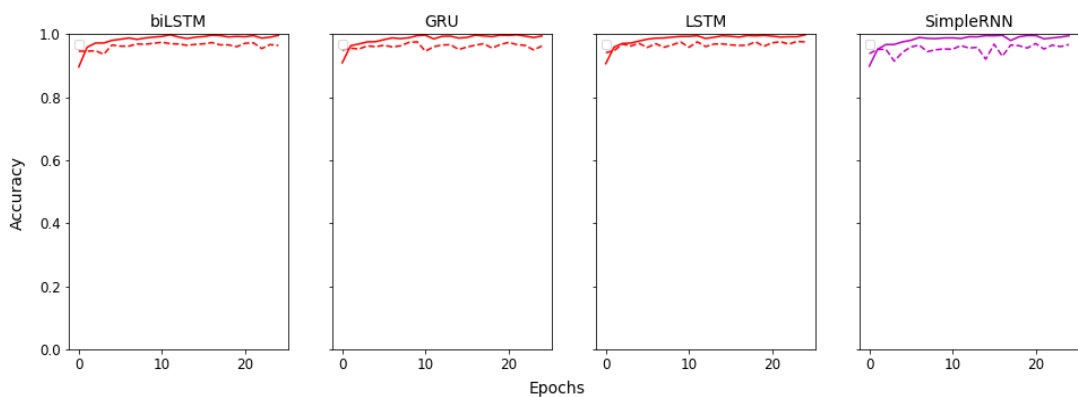


Figure 40 – Model Accuracy plots of the 32 cell RNN models

After passing the initial stages of training, an analysis was made on the results obtained. Recognition accuracy plots (train and test) were "non-representative" in the SimpleRNN model plot, figure 40, despite the fact that the average recognition accuracy scores were similar. In addition, the CNN network showed a lower recognition rate than other models. This suggests that the recognition performance of CNN models for the sounds of UAVs and other objects. The CNN layer can have high recognition capability if more CNN layers are added deeply. Two CNN layers were added because the CNN structure was based on previous work [1, p. 864]. However, compared to single-layer RNN models, the recognition of the CNN model was significantly lower as seen (table 7). At the same time, at least 2-3 attempts were made to repeatedly check each experiment. This is due to the assumption that a model trained only once can be a random chance of prediction. The GRU and SimpleRNN models were found to be significantly more accurate than the LSTM and BiLSTM models. The recognition history plot of the SimpleRNN network was found to be unrepresentative and was not continued in further studies. The next step of the study involved increasing the number of GRU cells to 64 and continuing training with 25 epochs. But the value of the "activity regularization" function in the penultimate layer was sought from a different interval due to an increase in the number of cells. In the GRU model with 64 cells, the values of the "activity regularization" function were

taken equal to $L2 = 0.00001$, which provided “good fit” to the recognition accuracy plot. (figure 41) shows the overall accuracy of the models CNN and GRU.

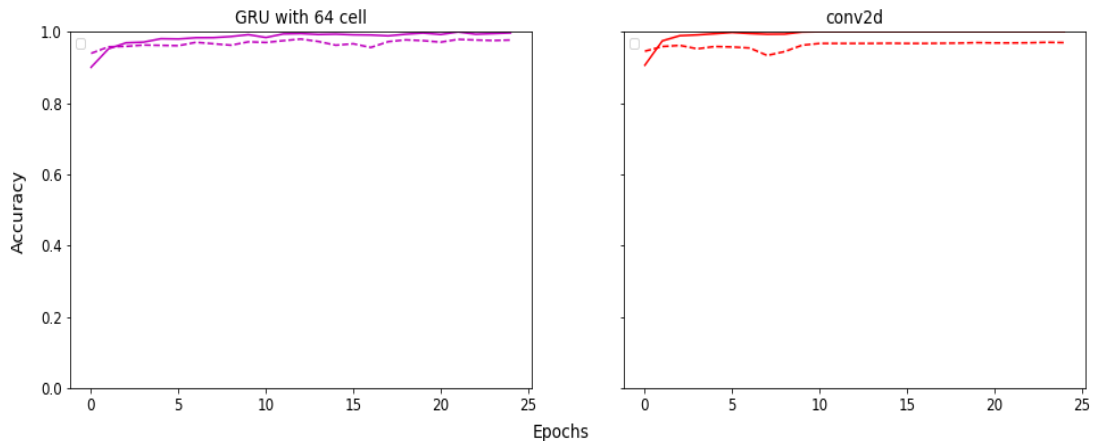


Figure 41 – Model Accuracy plots of the GRU model and CNN model

The proposed GRU model with 64 cells therefore provides a relatively good recognition ability, as illustrated in figure 25 above. Also, it displayed a "good fit" model accuracy plot. The CNN model exhibits an unrepresentative gap between training and testing accuracy, as well as a lower recognition capability than one-layer RNN architecture. In general, there was also a 4-class dataset performed during model building and testing. They were checked for drone sounds recorded in the immediate area. Also, these were performed on the basis of the first database with a small composition. A series of results from such a study is presented as a confusion matrix in the (Appendix I).

In general, it is impossible to accurately demonstrate the capabilities of a recognition system based on average recognition accuracy. Therefore, by presenting the recognition results in an extended form with recognition accuracy characteristics for each class, it is possible confidently assess the predictive power of the models.

The performance and robustness of the model in the case of many classification problems is usually assessed using the classifier confusion matrix. Sensitivity (recall), specificity and accuracy can be calculated using the components of the matrix. Many performance indicators [7, p. 3856(17-19)], including Precision, Recall and F1, were used to evaluate our strategy.

By calculating the ratio of false positive (FP) objects to true positive (TP) objects using equation (19), the accuracy can be determined as follows:

$$Precision = \frac{T_p}{T_p + T_p} \quad (19)$$

Equation (20) was used to evaluate recall by comparing true positive (TP) predictions with false negative (FN) predictions:

$$Recall = \frac{T_p}{T_p + F_p} \quad (20)$$

The F1 score, which reflects the average of the data, was calculated using equation (9), (10) because precision or recall does not properly assess system predictability, equation (22):

$$F_1 = \frac{2*Precision*Recall}{Precision+Recall} \quad (21)$$

Each class was made up of 100 files so that they could be visually seen when these scoring methods were used on the prediction dataset (300 files). The prediction results are shown in Table 8. A confusion matrix was also performed for the studied basic 5 models and the later developed 6th GRU model. The confusion matrix, which identifies and predicts each class, was able to provide sufficient information to predict the reliability of the given models (table 8).

Table 8 – The performance of the models and their Prediction Metric

Model	Classes	Precision, %	Recall, %	F1-score, %
Simple RNN with 32 cells	Background noise	99	100	100
	Loaded UAV	100	96	98
	Unloaded UAV	96	99	98
LSTM with 32 cells	Background noise	96	100	98
	Loaded UAV	98	97	97
	Unloaded UAV	98	95	96
BiLSTM with 32 cells	Background noise	97	100	99
	Loaded UAV	99	95	97
	Unloaded UAV	96	97	97
GRU with 32 cells	Background noise	99	99	99
	Loaded UAV	97	98	98
	Unloaded UAV	97	96	96
CNN as in [1]	Background noise	99	95	97
	Loaded UAV	95	89	92
	Unloaded UAV	90	99	94
GRU with 64 cells	Background noise	99	99	99
	Loaded UAV	99	98	98
	Unloaded UAV	97	98	98

As a result, when estimating the background noise class, almost all types of RNN models have a very high identifying ability. The CNN model also performed slightly worse than the RNN models, but had better accuracy in the background noise class. This demonstrates that while CNN models are capable of solving common recognition problems, they are less efficient than RNNs when dealing with the same sounds of objects that are in different states.

Moreover, almost all RNN cells have strong recognition abilities from a single layer and have a great ability to identify elements based on the engine. Along with the capabilities of the RNN network, the structure of the model also plays a special role in this situation. Table 7 shows that the "tanh" activation function was used to set

the dense layer prior to the RNN model. The dense layers also got the "relu" feature after the RNN layer.

To avoid overtraining, a Dropout layer has also been included. Also, the dataset was upgraded from its previous version. In this way, ideal recognition results were achieved. In both loaded and unloaded UAV situations, simple RNNs, LSTMs, and BiLSTMs failed to demonstrate consistently high sound recognition rates, as evidenced by the study of RNN models in table 8. True Positive recognition results, shows that the level of recognition of loaded and unloaded UAVs decreased by 4-5%. Also, the GRU with 32 cells showed the best performance for the main target class of loaded UAVs. And by expanding to 64 cells, the best results were achieved for all classes. This leads to the conclusion that GRU cells well and reliably recognize different noise states of the same object. And CNN models have proven to be effective in processing binary classification with a large number of levels. However, all varieties of RNNs have outperformed CNNs on binary and multiple sound classification problems due to their stable recognition capabilities.

This study concludes that the GRU model is a useful tool for recognizing UAV acoustic data in different states. The confusion matrix created using the 64 cells of the GRU model is shown in figure 42.

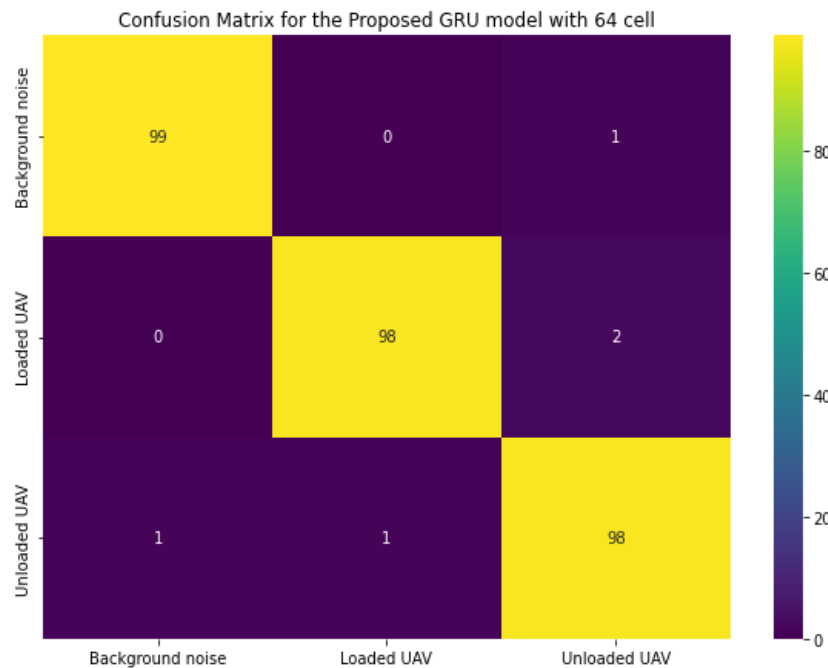


Figure 42 – The confusion matrix produced with the 64 GRU model cells

To sum up, the confusion matrices were designed to display accurately predicted and mis predicted audio files of UAV states. The proposed model has received wide recognition in the "Loaded UAV" and "Unloaded UAV" classes, which were the main focus. The ROC plot of the proposed model is shown in figure 27 to demonstrate the robustness of the model. To demonstrate the model's capacity for accurate performance on the given dataset, ROC curves were also obtained,

(figure 43). Background noise class was paraphrased as ambient noise in this assessing plot of the model.

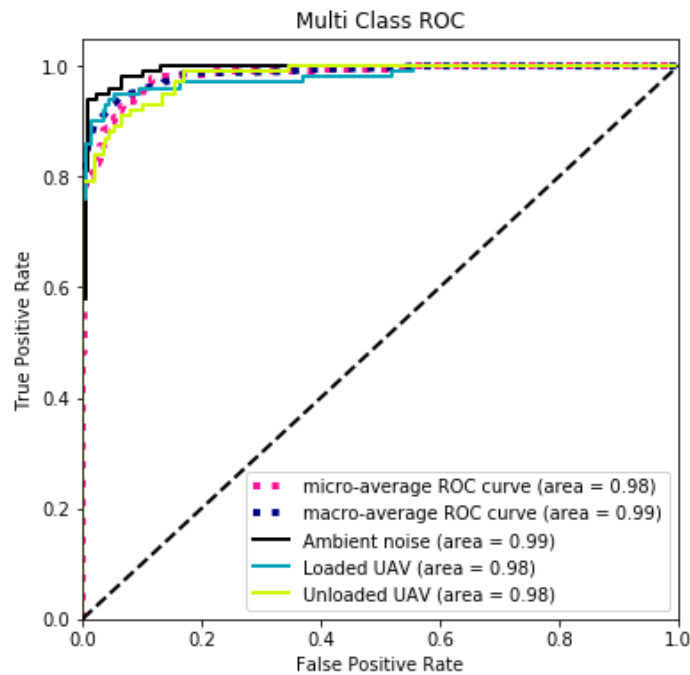


Figure 43 – ROC curve demonstrating model performance

As a result, the classifier received a data set divided into three classes. In order to evaluate the predictability of the chosen classifier and ensure that each distinct class is correctly scored, the actual predicted audio files and false negatives of the chosen classifier were mapped using a confusion matrix utilizing balanced 100 audio recordings per class. The results of the tests using the 64-cell GRU model clearly showed that the recognition skills based on the provided database were stable.

CONCLUSION

The main goal of this dissertation study was to solve the problem of UAV acoustic data recognition. In the course of realizing this goal, the SimpleRNN, LSTM, bidirectional LSTM and GRU architecture models have been explored in depth for the real-time UAV acoustic data recognition system. This work especially carefully examined the situation of whether the UAVs were loaded or unloaded. Unloaded UAVs, loaded UAVs, and background noises such as the sound of other engine-based objects from the background were the three main classes. Then, an efficient method for recognizing UAV acoustic data was determined. During the experiments (Appendix I, K, L), the accuracy of the UAV recognition system was evaluated using all the metrics from numerous class classification problems. As a result, the GRU architecture (64) was found to be an efficient model with a high level of predictability on the given dataset. In addition, this model can identify loaded and unloaded UAVs with 98% accuracy, as well as background noise with 99% accuracy. This evaluation confirms the reliability of the UAV audio recognition system and proposes to build a network of acoustic sensors using the proposed GRU model (64). Moreover, various RNN network architectures are robust to binary and multi-classification problems. Because they are better at content-based recognition than CNN models. To sum up, the SimpleRNN, LSTM, BiLSTM, and GRU networks with the proposed architecture can be used in the task of UAV load detection. The CNN model had a somewhat lower level of multiple classification on sounds than the RNN model. The CNN could better recognize binary classification instances as seen from experimental studies.

A limitation of this work is the smaller amount of acoustic data from loaded UAVs. However, this study showed that it is possible to recognize and evaluate UAV loads in real-time mode. A further continuation of the study took the direction of a bimodal method for detecting UAVs using software-defined radio (SDR). As a scientific continuation of this work, project “AP14971907” is being implemented, combining acoustic sensor and SDR methods, Appendix M. The system, in this study, is proposed as a scientific solution for small territorial-strategic areas and a bimodal method for ensuring national security. And for strategic areas with a large area, this acoustic sensor can be repeatedly placed at several points or nodes and carry out protection measures with centralized control.

REFERENCES

- 1 Li S., Kim H., Lee S.D. et al. Convolutional Neural Networks for Analyzing Unmanned Aerial Vehicles Sound // *Proc. 18th internat. conf. on Control, Automation, and Systems (ICCAS)*. – Daegu, 2018. – P. 862-866.
- 2 Lim D., Kim H., Hong S. et al. Practically Classifying Unmanned Aerial Vehicles Sound Using Convolutional Neural Networks // *Proc. 2nd IEEE internat. conf. on Robotic Computing (IRC)*. – Laguna Hills, 2018. – P. 242-245.
- 3 Vemula H.C. Multiple Drone Detection and Acoustic Scene Classification with Deep Learning. – Dayton: Wright State University, 2018. – 149 p.
- 4 Kim J., Park C., Ahn J. et al. Real-time UAV sound detection and analysis system // *Proc. IEEE Sensors Applications sympos. (SAS)*. – Glassboro, 2017. – P. 1-5.
- 5 Park S., Shin S., Kim Y. et al. Combination of Radar and Audio Sensors for Identification of Rotor-type Unmanned Aerial Vehicles (UAVs) // *Proc. 2015 IEEE Sensors*. – Busan, 2015. – P. 1-4.
- 6 Taha B., Shoufan A. Machine Learning-Based Drone Detection and Classification: State-of-the-Art in Research // *Proc. IEEE Access*. – 2019. – Vol. 7. – P. 138669-138682.
- 7 Seidaliyeva U., Akhmetov D., Ilipbayeva L. et al. Real-Time and Accurate Detection in a Video with a Static Background // *Sensors*. – 2020. – Vol. 20, Issue 14. – P. 3856-1-3856-18.
- 8 Косенов А. Казахстан подтвердил проникновение узбекского беспилотника на свою территорию // <https://tengrinews.kz/events/kazakhstan-podtverdil-proniknovenie-uzbekskogo-bespilotnika>. 20.12.2022.
- 9 Беспилотник захвачен над зданием Минобороны Казахстана // https://tengrinews.kz/kazakhstan_news/bespilotnik-zahvachen-nad. 20.12.2022.
- 10 Presechen nesanktsionirovannyyu polet kvadrokoptera nad zdaniyem Minoborony // <https://www.egemen.kz/article/201579-presechen>. 20.12.2022.
- 11 Военные перехватили дрон над Арысью // <https://ru.sputnik.kz/20190722/voennye-perekhvatili-dron-nad-arysyu-11014334.html>. 20.12.2022.
- 12 Houthi drone crashes into Saudi school in Asir province // <https://thearabweekly.com/houthi-drone-crashes-saudi-school-asir-province>. 20.12.2022.
- 13 Mircea Cr. Light Show in China May Have Been Sabotaged, Dozens of Drones Fell From the Sky // <https://www.autoevolution.com/news/light>. 20.11.2020.
- 14 Why Drones Are Becoming More Popular Each Day // <https://www.entrepreneurshipinbox.com/22657/why-drones-are-becoming>. 20.11.2020.
- 15 Utebayeva D., Almagambetov A., Alduraibi M. et al. Multi-label UAV sound classification using Stacked Bidirectional LSTM // *Proc. 4th internat. conf. on Robotic Computing (IRC)*. – Taichung, 2020. – P. 453-458.
- 16 Utebayeva D., Alduraibi M., Ilipbayeva L. et al. Stacked BiLSTM - CNN for Multiple label UAV sound classification // *Proc. 4th internat. conf. on Robotic Computing (IRC)*. – Taichung, 2020. – P. 470-474.

- 17 McFarland M. Airports Scramble to Handle Drone Incidents // <https://www.cnn.com/2019/03/05/tech/airports-drones/index.html>. 15.08.2021.
- 18 Li J., Dai W., Metze F. et al. A comparison of Deep Learning methods for environmental sound detection // 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). – New Orleans, 2017. – P. 126-130.
- 19 Ranft R. Natural sound archives: past present and future // Anais da Academia Brasileira de Ciências. – 2004. – Vol. 76, Issue 2. – P. 455-465.
- 20 Utebayeva D., Ilipbayeva L., Matson E.T. Practical Study of Recurrent Neural Networks for Efficient Real-Time Drone Sound Detection: A Review // Drones. – 2023. – Vol. 7, Issue 1. – P. 26-1-26-25.
- 21 Drone Crash Database / <https://dronewars.net/drone-crash>. 04.02.2021.
- 22 Koslowski R., Schulzke M. Drones along Borders: Border Security UAVs in the United States and the European Union // <https://www.albany.edu> s. 04.02.2021.
- 23 Shi W., Arabadjis G., Bishop B. et al. Detecting, Tracking, and Identifying Airborne Threats with Netted Sensor Fence // In book: Sensor Fusion Foundation and Applications: In Tech, 2011. – 238 p.
- 24 Samaras S., Diamantidou E., Ataloglou D. et al. Deep Learning on Multi Sensor Data for Counter UAV Applications – A Systematic Review // Sensors. – 2019. – Vol. 19. – P. 4837-1-4837-35.
- 25 Ezuma M., Erden F., Anjinappa C.K. et al. Micro-UAV Detection and Classification from RF Fingerprints Using Machine Learning Techniques // Proceed. of the IEEE AERO. – Big Sky, MT, USA, 2019. – P. 1-13.
- 26 Jeon S., Shin J.W., Lee Y.J. et al. Empirical Study of Drone Sound Detection in Real-Life Environment with Deep Neural Networks. 25th European Signal Processing conf. (EUSIPCO). – Kos, 2018. – P. 1858-1862.
- 27 Delivery drone crashes into power lines, causes outage // https://www.theregister.com/2022/09/30/delivery_drone_crashes_into. 25.08.2021.
- 28 When Amazon drones crashed, the company told the FAA to go fly a kite // <https://www.businessinsider.com/amazon-prime-air-faa-regulators>. 25.08.2021.
- 29 List of unmanned aerial vehicles-related incidents // https://en.wikipedia.org/wiki/List_of_unmanned_aerial_vehicles-related. 25.08.2021.
- 30 Amazon Drone Project Layoffs // <https://www.idtechex.com/en/research-article/amazon-drone-project-layoffs/25951>. 25.08.2021.
- 31 Utebayeva D. Effectiveness of the system of unmanned aerial vehicles detection on the basis of acoustic signature // Vestnik KazNRTU. – 2020. – Vol. 4, Issue 140. – P. 300-307.
- 32 Al-Emadi S. et al. Audio Based Drone Detection and Identification using Deep Learning // Proceed. 15th internat. Wireless Communications & Mobile Computing conf. (IWCMC). – Tangier, 2019. – P. 459-464.
- 33 Shi L., Ahmad I., He Y.J. et al. Hidden Markov Model based Drone Sound Recognition using MFCC Technique in Practical Noisy Environments // Journal of Communications and Networks. – 2018. – Vol. 20, Issue 5. – P. 509-518.

- 34 Siriphun N., Kashihara S. et al. Distinguishing Drone Types Based on Acoustic Wave by IoT Device // *Proced., 22nd internat. Computer Science and Engineering conf. (ICSEC)*. – Chiang Mai, 2018. – P. 1-4.
- 35 Anwar M.Z., Kaleem Z., Jamalipour A. Machine Learning Inspired Sound-Based Amateur Drone Detection for Public Safety Applications // *IEEE Transactions on Vehicular Technology*. – 2019. – Vol. 68, Issue 3. – P. 2526-2534.
- 36 Liu H., Wei Zh., Chen Y. et al. Drone Detection based on An Audio-assisted Camera Array // *Proced. IEEE 3rd internat. conf. on Multimedia Big Data*. – Laguna Hills, CA, USA, 2017. – P. 402-406.
- 37 Bernardini A., Mangiatordi F., Pallotti E. et al. Drone detection by acoustic signature identification // *Electronic Imaging*. – 2017. – Issue 10. – P. 60-64.
- 38 Shaikh F. Getting Started with Audio Data Analysis using Deep Learning // <https://www.analyticsvidhya.com/blog/2017/08/audio-voice-processing>. 14.03.2021.
- 39 Sahidullah M., Saha G. Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition // *Speech Communication*. – 2012. – Vol. 54, Issue 4. – P. 543-565.
- 40 Wang Y., Fagian F.E., Ho K.E. et al. A Feature Engineering Focused System for Acoustic UAV Detection // *Proced. 5th IEEE internat. conf. on Robotic Computing (IRC)*. – Taichung, 2021. – P. 125-130.
- 41 Kim B., Jang B., Lee D. et al. CNN-based UAV Detection with Short Time Fourier Transformed Acoustic Features // *Proced. internat. conf. on Electronics, Information, and Communication (ICEIC)*. – Barcelona, 2020. – P. 1-3.
- 42 Z. Shi, L. Zheng, X. Zhang, Y. Wang and L. Wu, "CNN-Based Electronic Camouflage Audio Restoration Mechanism Zhengyu Shi," 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, China, 2018, pp. 412-416.
- 43 Choi K., Joo D., Kim J. Kapre: On-GPU Audio Preprocessing Layers for a Quick Implementation of Deep Neural Network Models with Keras // <https://doi.org/10.48550/arXiv.1706.05781>. 10.04.2021.
- 44 Nijim M., Mantrawadi N. Drone classification and identification system by phenome analysis using data mining techniques // *Proced. IEEE sympos. on Technologies for Homeland Security (HST)*. – Waltham, MA, 2016. – P. 1-5.
- 45 Yue X., Liu Y., Wang J. et al. Software defined radio and wireless acoustic networking for amateur drone surveillance // *IEEE Commun. Mag.* – 2018. – Vol. 56, Issue 4. – P. 90-97.
- 46 Seo Y., Jang B., Im S. Drone detection using convolutional neural networks with acoustic STFT features // *Proced. IEEE internat. conf. on Advanced Video and Signal Based Surveillance (AVSS)*. – Auckland, 2018. – P. 1-6.
- 47 Matson E., Yang B., Smith A. et al. UAV detection system with multiple acoustic nodes using machine learning models // *Proced. 3rd IEEE internat. conf. on Robotic Computing (IRC)*. – Naples, 2019. – P. 493-498.
- 48 Утебаева Д., Илипбаева Л. Әр түрлі модельдер үшін ұшқышсыз әуе көліктерін анықтау мәселелерінде акустикалық сигналдарды зерттеу // *Вестник АУЭС*. – 2020. – №3(50). – Б. 38-46.

- 49 Salehinejad H., Sankar S., Barfett J. et al. Recent Advances in Recurrent Neural Networks // <https://arxiv.org/pdf/1801.01078.pdf>. 22.02.2018.
- 50 Sak H., Senior A.W., Senior A.W. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling // *Proceed. conf. of the internat. Speech Communication Association. – Thailand and Penang (Malaysia), 2014. – P. 338-342.*
- 51 Brownlee J. The Long Short-Term Memory Network // *In book: Long Short-Term Memory Networks with Python. – USA, 2019. – P. 10-12.*
- 52 Thakur D. LSTM and its equations // <https://medium.com/@divyanshu132/lstm-and-its-equations-5ee9246d04af>. 10.11.2019.
- 53 Cho K., Van Merriënboer B., Gulcehre C. et al. Learning phrase representations using rnn encoder-decoder for statistical machine translation // *Proceed. of the conf. on Empirical Methods in Natural Language Processing (EMNLP) – Doha, 2014. P. – 1724-1734.*
- 54 Audio signal // https://en.wikipedia.org/wiki/Audio_signal. 10.05.2019.
- 55 Magalhães E., Jacob J., Nilsson N. et al. Physics-based Concatenative Sound Synthesis of Photogrammetric models for Aural and Haptic Feedback in Virtual Environments // *Proceed. IEEE conf. on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW). – Atlanta, GA, 2020. – P. 376-379.*
- 56 Sound Waves // <https://www.pasco.com/products/guides/sound>. 14.04.2019.
- 57 What is a Sound Wave in Physics? // <https://www.pasco.com>. 14.04.2019.
- 58 Sound // <https://en.wikipedia.org/wiki/Sound>. 14.04.2019.
- 59 Sine Wave: Definition, What It's Used For, Example, and Causes // <https://www.investopedia.com/terms/s/sinewave.asp>. 04.04.2022.
- 60 Sine wave // https://en.wikipedia.org/wiki/Sine_wave. 14.04.2019.
- 61 Discrete time and continuous time // <https://en.wikipedia.org/wiki>. 14.04.2019.
- 62 Introduction to Audio Signal Processing // <https://www.coursera.org/learn/audio-signal-processing/lecture/fHh1/introduction-to-audio>. 14.04.2019.
- 63 Sampling (signal processing) // <https://en.wikipedia.org/wiki>. 14.04.2019.
- 64 Wu Y., Krishnan S. Classification of knee-joint vibroarthrographic signals using time-domain and time-frequency domain features and least-squares support vector machine // *Proceed. 16th internat. conf. on Digital Signal Processing. – Santorini, Greece, 2009. – P. 1-6.*
- 65 Time series // https://en.wikipedia.org/wiki/Time_series. 25.08.2019.
- 66 Window function // <https://en.wikipedia.org/wiki/Window>. 25.08.2019.
- 67 Bahoura M. FPGA Implementation of Blue Whale Calls Classifier Using High-Level Programming Tool // *Electronics. – 2016. – Vol. 5. – P. 8-1-8-19.*
- 68 time_frequency // <https://kapre.readthedocs.io/en/latest/time>. 25.08.2019.
- 69 Types of Neural Networks: Applications, Pros, and Cons // <https://www.knowledgehut.com/blog/data-science/types-of-neural>. 25.08.2019.
- 70 Graves A., Mohamed A., Hinton G. Speech recognition with deep recurrent neural networks // *Proceed. IEEE internat. conf. on Acoustics, Speech and Signal Processing. – Vancouver, BC, 2013. – P. 6645-6649.*

71 Jie Zhao. Anomalous Sound Detection Based on Convolutional Neural Network and Mixed Features // Journal of Physics: Conference Series. – 2020. – Vol. 1621. – P. 1-8.

72 Afridi T.H., Alam A., Khan N. A Multimodal Memes Classification: A Survey and Open Research Issues // In book: Innovations in Smart Cities Applications. Cham: Springer, 2020. – Vol. 4. – P. 1451-1466.

73 What Is a Convolutional Neural Network? // <https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html>. 10.02.2019.

74 Shu H., Song Y., Zhou H. Time-frequency Performance Study on Urban Sound Classification with Convolutional Neural Network // Proc. IEEE Region 10 conf. (Tencon 2018). – Jeju (Korea), 2018. – P. 1713-1717.

75 Momo N., Abdullah, Uddin J. Speech Recognition Using Feed Forward Neural Network and Principle Component Analysis // Advances in Intelligent Systems and Computing: proced. internat. sympos. on Signal Processing and Intelligent Recognition Systems. – Manipal, 2018. – P. 228-239.

76 Segarceanu S., Suci G., Gavati I. Neural Networks for Automatic Environmental Sound Recognition // Proc. internat. conf. on Speech Technology and Human-Computer Dialogue (SpeD). – Bucharest, 2021. – P. 7-12.

77 Shamsuddin N., Mustafa M.N., Husin S. et al. Classification of heart sounds using a multilayer feed-forward neural network // Proc. Asian conf. on Sensors and the internat. conf. on New Techniques in Pharmaceutical and Biomedical Research. – Kuala Lumpur, 2005. – P. 87-90.

78 Main Types of Neural Networks and Its Applications – Tutorial. 13 July 2020 // <https://towardsai.net/p/machine-learning/main-types-of-neural>. 20.12.2022.

79 Mahyub M., Souza L.S., Batalo B. et al. Environmental Sound Classification Based on CNN Latent Subspaces // Proc. internat. Workshop on Acoustic Signal Enhancement (IWAENC). – Bamberg, 2022. – P. 1-5.

80 Wu Y., Lee T. Enhancing Sound Texture in CNN-based Acoustic Scene Classification // Proc. IEEE internat. conf. on Acoustics, Speech and Signal Processing (ICASSP-2019). – Brighton, 2019. – P. 815-819.

81 Wang Y., Chu Z., Ku I. et al. A Large-Scale UAV Audio Dataset and Audio-Based UAV Classification Using CNN // Proc. 6th IEEE internat. conf. on Robotic Computing (IRC). – Milan, 2022. – P. 186-189.

82 Bubashait M., Hewahi N. Urban Sound Classification Using DNN, CNN & LSTM a Comparative Approach // Proc. internat. conf. on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT). – Zallaq, Bahrain, 2021. – P. 46-50.

83 Noman F., Ting C.-M., Salleh S.-H. et al. Short-segment Heart Sound Classification Using an Ensemble of Deep Convolutional Neural Networks // Proc. IEEE internat. conf. on Acoustics, Speech and Signal Processing (ICASSP-2019). – Brighton, 2019. – P. 1318-1322.

84 Lu R., Duan Z., Zhang C. Multi-Scale Recurrent Neural Network for Sound Event Detection // Proc. IEEE internat. conf. on Acoustics, Speech and Signal Processing (ICASSP). – Calgary, AB, 2018. – P. 131-135.

85 Parascandolo G., Huttunen H., Virtanen T. Recurrent neural networks for polyphonic sound event detection in real life recordings // *Proceed. IEEE internat. conf. on Acoustics, Speech and Signal Processing (ICASSP)*. – Shanghai, 2016. – P. 6440-6444.

86 Semmad A., Bahoura M. Long Short Term Memory Based Recurrent Neural Network for Wheezing Detection in Pulmonary Sounds // *Proceed. IEEE internat. Midwest sympos. Circuits and Systems (MWSCAS)*. – Lansing, MI, 2021. – P. 412-415.

87 Kamepalli S., Rao B.S. et al. Multi-Class Classification and Prediction of Heart Sounds Using Stacked LSTM to Detect Heart Sound Abnormalities // *Proceed. 3rd internat. conf. for Emerging Technology (INCET)*. – Belgaum, 2022. – P. 1-6.

88 Feng Y., Liu Z.J., Ling Y. et al. A Two-Stage LSTM Based Approach for Voice Activity Detection with Sound Event Classification // *Proceed. IEEE internat. conf. on Consumer Electronics (ICCE)*. – Las Vegas, NV, 2022. – P. 1-6.

89 Wang Y., Liao W., Chang Y. Gated Recurrent Unit Network-Based Short-Term Photovoltaic Forecasting // *Energies*. – 2018. – Vol. 11. – P. 2163-1-2163-14.

90 Fan T., Zhu J., Cheng Y. et al. A New Direct Heart Sound Segmentation Approach using Bi-directional GRU // *Proceed. 24th internat. conf. on Automation and Computing (ICAC)*. – Newcastle Upon Tyne, UK, 2018. – P. 1-5.

91 Peng N. et al. Environment Sound Classification Based on Visual Multi-Feature Fusion and GRU-AWS // *IEEE Access*. – 2020. – Vol. 8. – P. 191100-191114.

92 Tsalera E.; Papadakis, A.; Samarakou, M. Comparison of Pre-Trained CNNs for Audio Classification Using Transfer Learning // *J. Sens. Actuator Netw.* – 2021. – Vol. 10. – P. 721-72-22.

APPENDIX A

Table 1 – Model and layers of the CNN algorithm based on the publication

№	Layers	# of layers	Number of filters for each layer
1_CNN_by_[1]	Convolutional (Conv2D) Activation functions: Relu MaxPooling (MaxPool2D) BatchNormalization	1	32 (kernel size (3,3), strides (2,2))
2_CNN_by_[1]	Convolutional (Conv2D) Activation functions: Relu MaxPooling (MaxPool2D) BatchNormalization	1	64 (kernel size (3,3), strides (2,2))
3_CNN_by_[1]	Flatten Fully Connected Neural Network Activation functions: <i>Relu</i> Dropout Fully Connected Neural Network Activation function: <i>softmax</i>	1 1 1	100 0.7 3
1_CNN_by_[2]	Convolutional (Conv2D) Activation functions: Relu MaxPooling (MaxPool2D) BatchNormalization	1	10 (kernel size (3,3), strides (2,2))
2_CNN_by_[2]	Flatten Fully Connected Neural Network Activation functions: <i>Sigmoid</i> Dropout Fully Connected Neural Network Activation function: <i>softmax</i>	1 1 1	10 0.1 3
Note – Compiled according to the source [48, p. 40]			

APPENDIX B

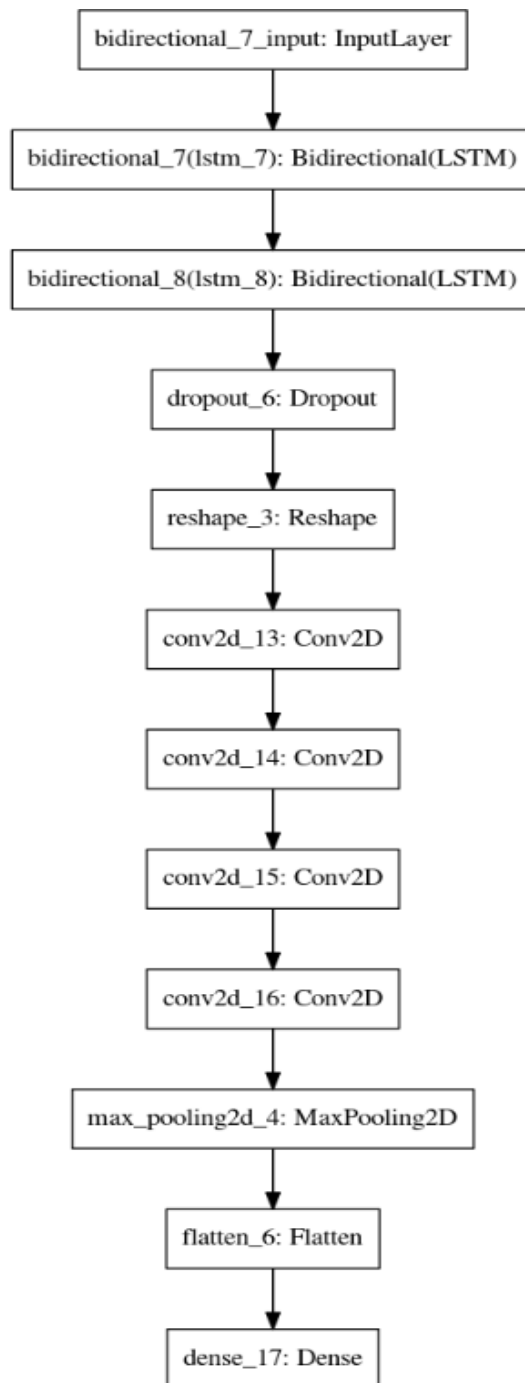


Figure B.1 – Visualization of the Stacked BiLSTM-CNN model presented in publication

Note – Compiled according to the source [16, p. 472]

APPENDIX C

		<i>Predicted</i>			
<i>True Actual</i>		<i>Loaded UAV</i>	<i>No UAV: ambient noise</i>	<i>No UAV: different background noises</i>	<i>Unloaded UAV</i>
	<i>Loaded UAV</i>	788	3	4	5
	<i>No UAV: ambient noise</i>	30	9280	1091	13
	<i>No UAV: different background noises</i>	5	307	1688	0
	<i>Unloaded UAV</i>	28	17	4	1334
	<i>SUPPORT</i>	800	10414	2000	1383

Figure C.1 – Composition of the initial dataset of the study

APPENDIX D

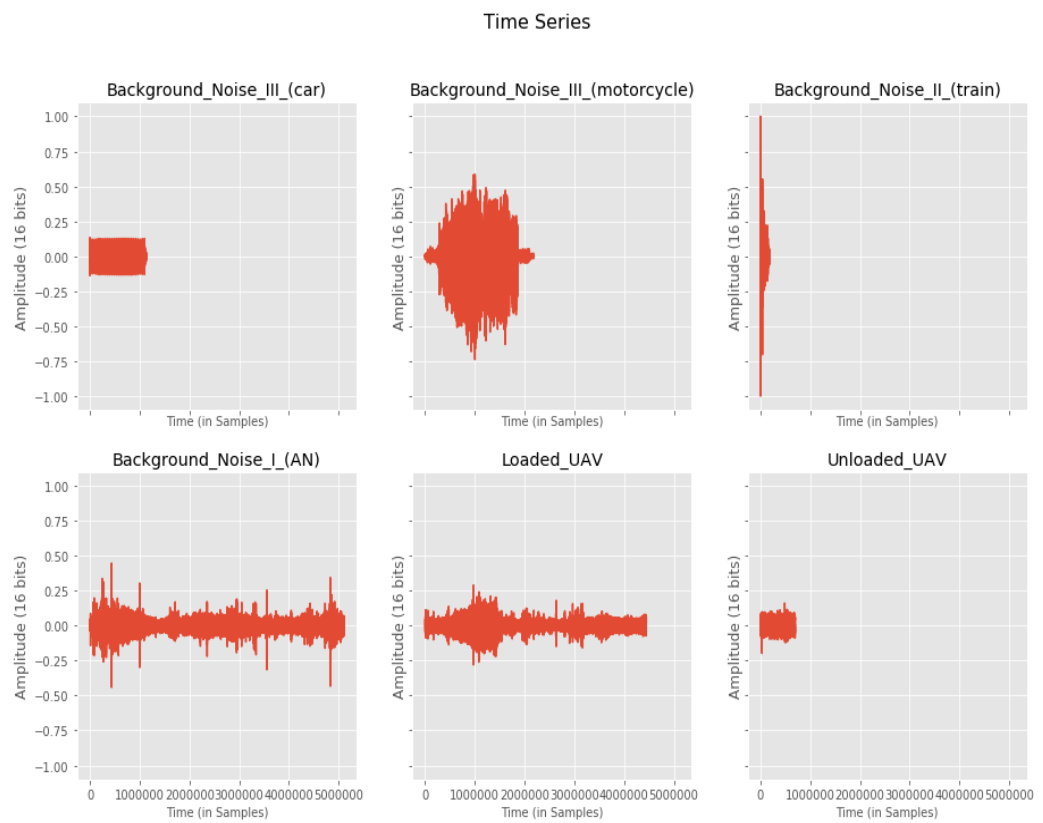
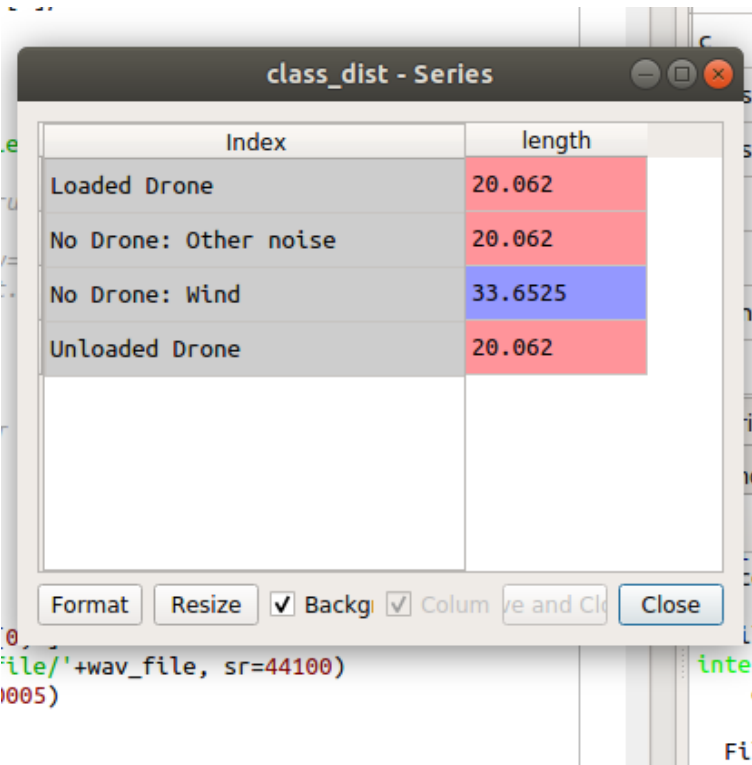
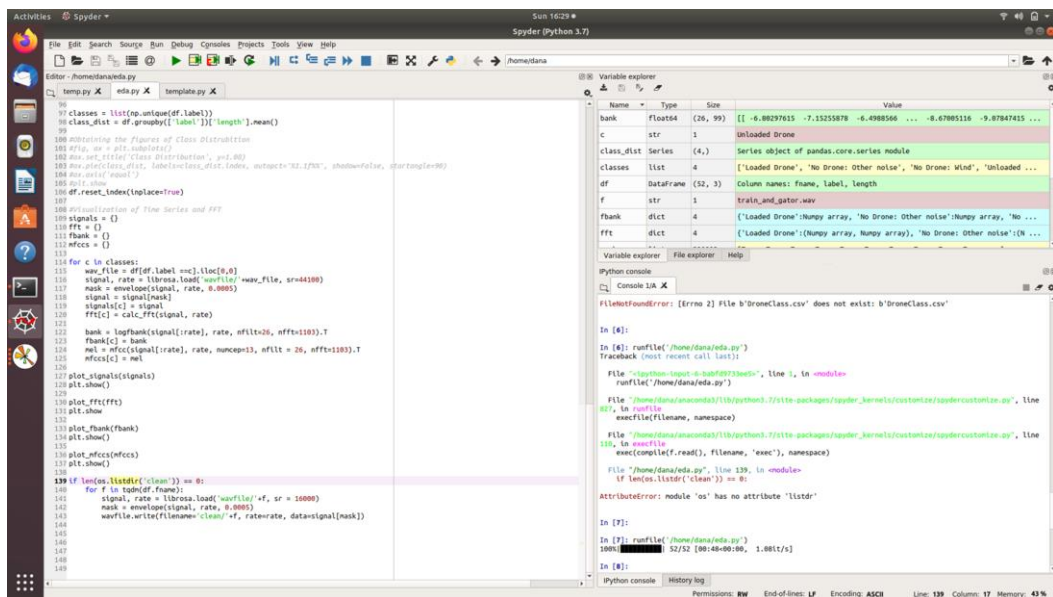


Figure D.1 – Studying the sounds of background objects and UAVs with 6 classes in the Time domain

APPENDIX E



a



b

a – audio recordings of various lengths in original length; b – the process of studying the filter bank, Melspectrogram and MFCC coefficients with various hyperparameters

Figure E.1 – Experimental studies at the stage of audio data adaptation

APPENDIX F

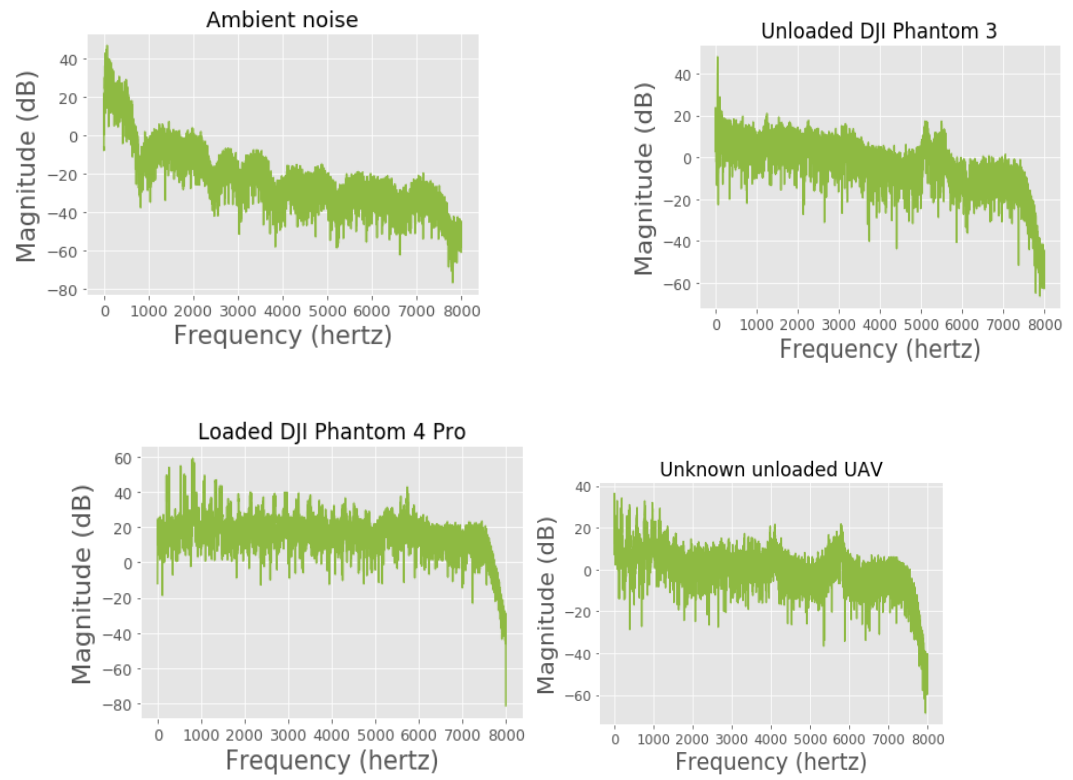
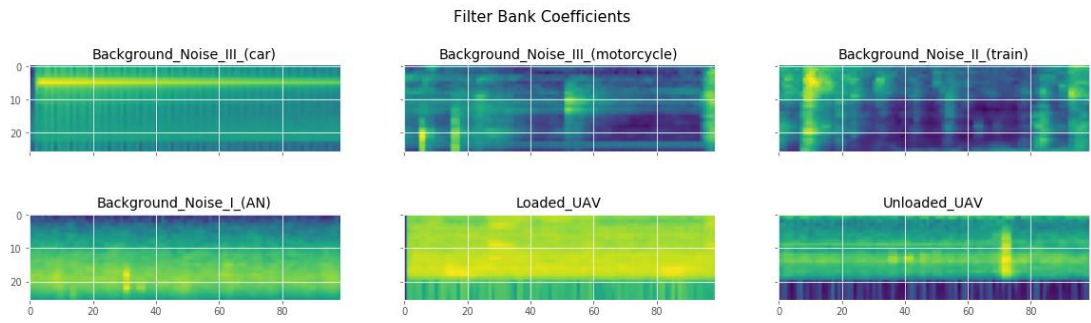


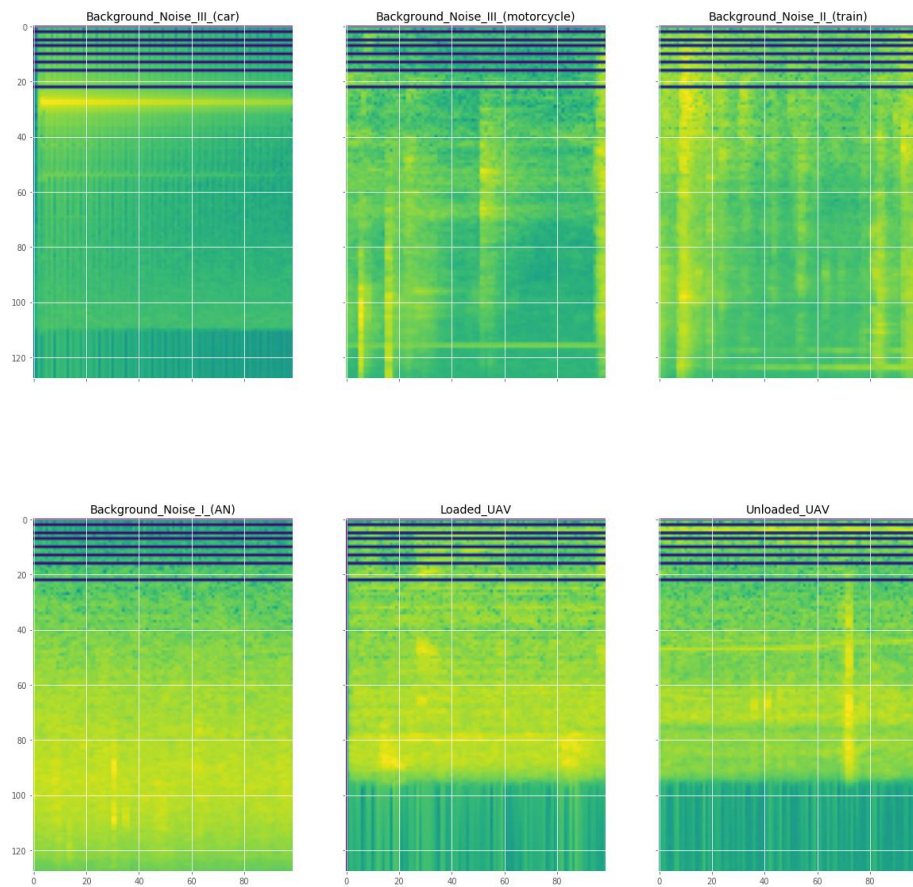
Figure F.1 – Plot of the Power level of UAV sound signals

APPENDIX G



a

Filter Bank Coefficients

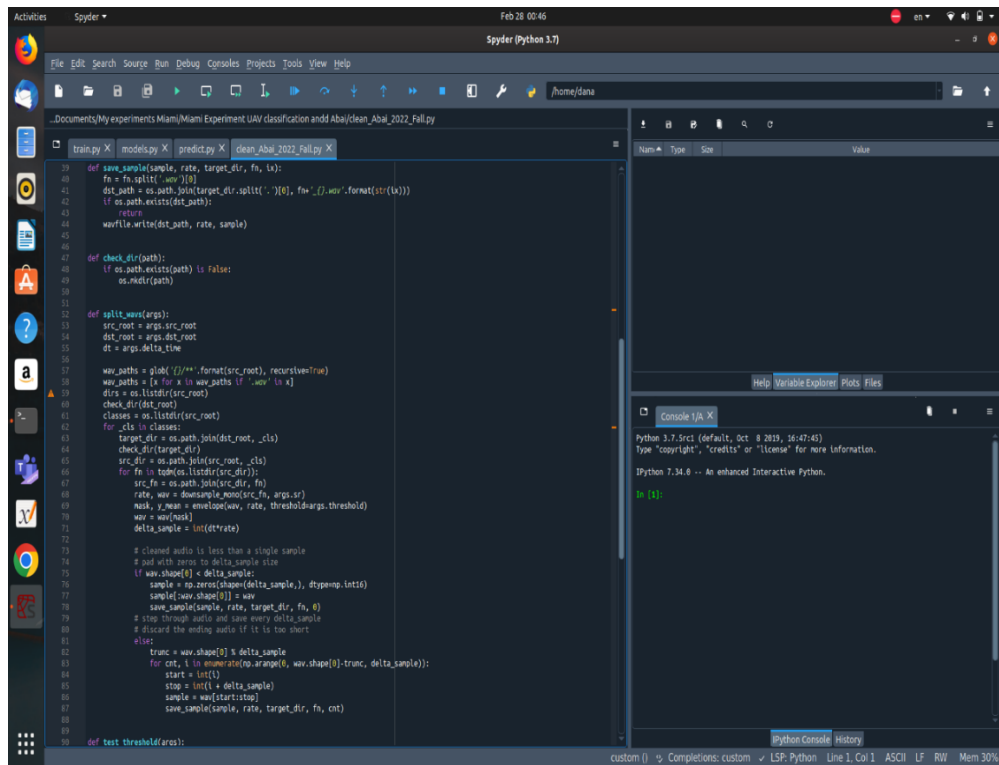


b

a – filterbank Coefficients for Class 6 UAV Sounds and Ambient Noises by the dimension 40 by 100; b – decimal Spectrograms for UAV sound signals of the proposed system

Figure G.1 – Investigation of spectrograms with various hyperparameters during the experiment

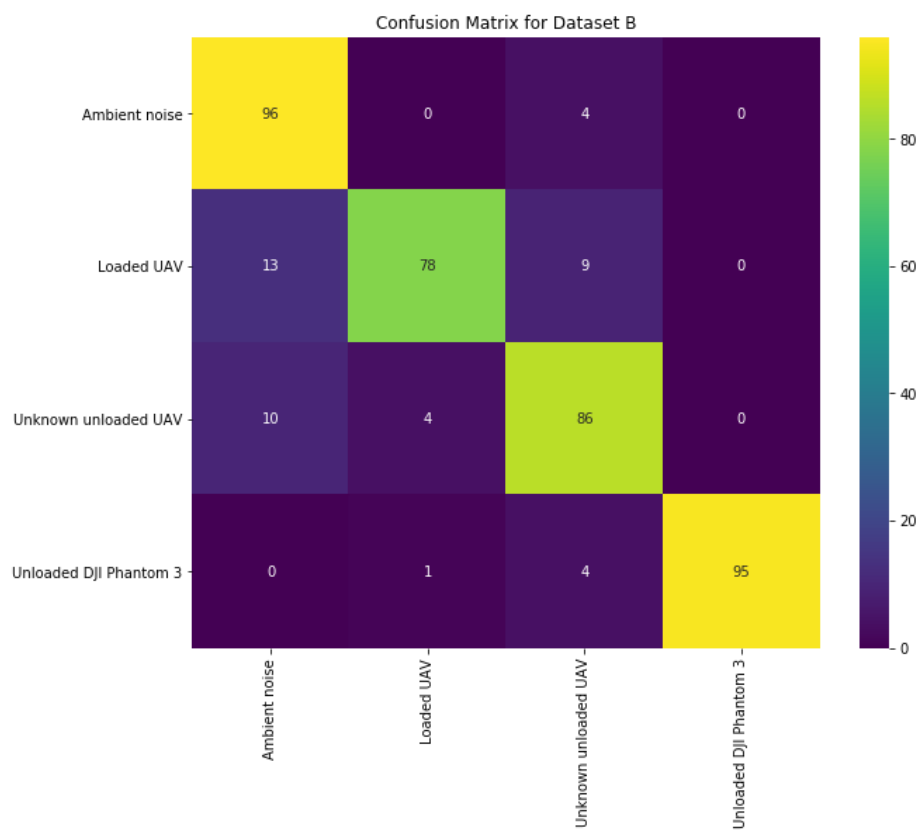
APPENDIX H



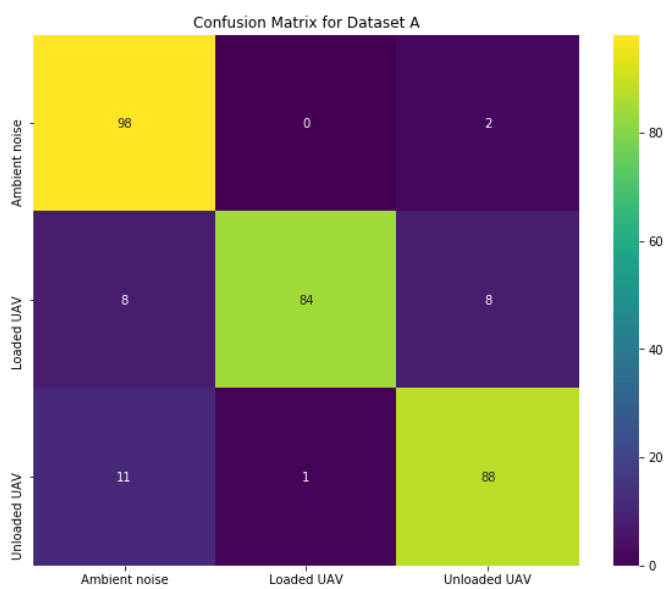
```
79 def save_sample(sample, rate, target_dir, fn, fs):
80     fn = fn.split('.')[-1]
81     dst_path = os.path.join(target_dir, split('.')[-1], fn+'.wav'.format(str(fs)))
82     if os.path.exists(dst_path):
83         return
84     wavfile.write(dst_path, rate, sample)
85
86
87 def check_dir(path):
88     if os.path.exists(path) is False:
89         os.mkdir(path)
90
91
92 def split_wav(args):
93     src_root = args.src_root
94     dst_root = args.dst_root
95     dt = args.delta_time
96
97     wav_paths = glob.glob('/**/*.wav', recursive=True)
98     wav_paths = [x for x in wav_paths if 'wav' in x]
99     dirs = os.listdir(src_root)
100     check_dir(dst_root)
101     classes = os.listdir(src_root)
102     for _cls in classes:
103         target_dir = os.path.join(dst_root, _cls)
104         check_dir(target_dir)
105         src_dir = os.path.join(src_root, _cls)
106         for fn in tqdm(os.listdir(src_dir)):
107             src_fn = os.path.join(src_dir, fn)
108             rate, wav = downsample_mono(src_fn, args.dt)
109             mask, y_max = envelope(wav, rate, threshold=args.threshold)
110             wav = wav[mask]
111             delta_sample = int(dt*rate)
112
113             # cleaned audio is less than a single sample
114             # pad with zeros to delta_sample size
115             if wav.shape[0] < delta_sample:
116                 sample = np.zeros(shape=(delta_sample, ), dtype=np.int16)
117                 sample[:wav.shape[0]] = wav
118             else:
119                 start = int(1)
120                 stop = int(1 + delta_sample)
121                 sample = wav[start:stop]
122                 save_sample(sample, rate, target_dir, fn, cnt)
123
124
125 def test_threshold(args):
```

Figure H.1 – Implementation of the proposed system in the Python program

APPENDIX I



a



b

a – 4-class database recognition experiment; b – 3-class database recognition experiment

Figure I.1 – Confusion matrix from an experiment on recognizing UAVs at close range and a certain state

APPENDIX K

Conducting experimental studies at international research institutions



Ronald A. Madler, Ph.D.
madler@erau.edu
Dean
College of Engineering
T: 928-777-3896

January 19, 2019

Dear Dana Utebayeva,

It is my pleasure to invite you to the Prescott Arizona Campus of Embry-Riddle Aeronautical University to work with Dr. Akhan Almagambetov of the Computer, Electrical and Software Engineering Department. The university is processing an exchange research/scholar J-visa for you. Prof. Almagambetov will be working with you to engage in cultural activities on and off campus. You will participate in the International Student Festival on February 23, 2019, which will showcase international food, activities, and performances by the international student body at Embry-Riddle. This festival attracts over 600 visitors from campus and the local community. In addition, several cultural trips to Arizona landmarks are planned while the students are on campus.

You will work under Dr. Almagambetov's direction during the period of 9 January 2019 through 31 July 2019 to complete your research topics. Research on automatic control for preservation of laminar flow via computer vision methods. Current flow visualization methods, for the most part, use manual methods of fluid dynamics analysis. This research will attempt to automate some aspects of this process, in effect providing users with a real-time computational fluid dynamics (CFD) capability during testing. Instantaneous feedback will be used to correct any flow anomalies, leading to laminar flow.

Your dissertation topic, "Efficient methods for object classification using deep learning," closely aligns with the research currently being performed at Embry-Riddle. This research will attempt to classify flow patterns using deep learning and artificial neural networks, thereby automating some aspects of flow visualization during wind-tunnel testing, with the goal of providing real-time feedback for the correction of turbulent flow. Embry-Riddle currently has state-of-the-art wind tunnel testing facilities, which will be instrumental in completing this research.

In addition to the generous financial support of your university, Dr. Almagambetov has additional financial resources to cover any shortfalls that may occur in your housing and living expenses. We thank you for your understanding as our university learns how to process the visiting exchange researcher J-visa.

Sincerely,

A handwritten signature in black ink that reads "Ronald A. Madler".

Ronald A. Madler
Dean, College of Engineering



3700 Willow Creek Road
Prescott, AZ 86301-3720

February 12, 2020

Dana Utebayeva
KazNRTU named after K.I.Satpayev
22,a, Satpayev Street
Almaty
Kazakhstan

Re: Offer to Extend Appointment as Visiting Scholar at Purdue University

Dear Ms. Utebayeva:

On behalf of Dean Bertoline and the Purdue Polytechnic Institute, it is my sincere pleasure to extend your appointment as a Visiting Scholar in the Computer and Information Technology Department at Purdue University April 30, 2020 through August 31, 2020. This offer is contingent upon the satisfaction of various conditions as described in this letter.

Visiting Scholars are invited to the University to engage in scholarly activities for their own academic enrichment and that of the department in which they have an appointment. Professor Eric Matson will serve as your principal point of contact while you are at Purdue University. Although you will have no formal departmental duties, we hope that you will become an active member of our scholarly community and will participate in University events. It is expected that you will work on CUAUV research for design and analysis of ML for acoustic sensors while at Purdue.

Applicable Terms & Conditions affecting Visiting Scholars

Your Visiting Scholar appointment does not carry any salary or benefits. Purdue University will continue to provide you a \$1400 per month living allowance for the additional 4 months (May 2020 – August 2020). You will be eligible to purchase a parking permit during the length of your appointment, but prior to leaving the University, we ask that you return your permit to Parking Facilities. The permit is non-transferable. In addition, you will be issued a Purdue identification card, be able to use library facilities, and your name will be listed in the University directory and on appropriate mailing lists.

As a Visiting Scholar at Purdue University, your appointment is subject to all applicable Purdue University policies, as they may be amended from time to time. It is your responsibility to become acquainted with the following policies, which are specifically incorporated into this letter:

1. C-12 "Classes of Purdue University Appointments for Personnel Not on the University Payroll"
<http://www.purdue.edu/policies/human-resources/c-12.html>

2. I.A.1 "Intellectual Property"

www.purdue.edu/policies/academic-research-affairs/ia1.html

Please note that policy I.A.1 referenced above requires Visiting Scholars who create intellectual property ("IP") in the course of their appointment with Purdue University to execute a general assignment of such IP in favor of Purdue, subject to certain exceptions, including one for certain scholarly and instructional copyrightable works. By accepting this offer letter, you will be making a prospective assignment of Purdue Intellectual Property (as defined in policy I.A.1) that you create in the course of your appointment by the University.

Conditional Offer

This offer is also contingent upon your obtaining and maintaining appropriate immigration status to permit you to work as a Visiting Scholar.

This letter and the policies referenced above contain the entire agreement concerning your appointment with the University. If these terms are acceptable and if you assent to the assignment of Purdue Intellectual Property, as described above and defined in Policy I.A.1, please sign where indicated below and return a signed copy to Misty Clugh, mclugh@purdue.edu at your earliest convenience.

The faculty and staff join me in welcoming you to Purdue University and look forward to continue working with you. We trust that it will be mutually rewarding.

Sincerely,



Robert F. Cox, Ph.D.
Senior Associate Dean for Globalization
Purdue Global Fellow, Office of Corporate and Global Partnerships
Professor of Construction Management Technology

RFC:mnc

APPENDIX L

Publication of experimental studies at the conference



The Fourth IEEE International Conference on Robotic Computing (IEEE IRC 2020)

Invitation Letter

February 4, 2020

Dear author / Dana Utebayeva,

On behalf of the Fourth IEEE International Conference on Robotic Computing (IEEE IRC 2020), it is our pleasure to inform you that your paper submitted to the **6th International Workshop on Collaboration for Humans, Agents, Robots, Machines and Sensors (CHARMS 2020)**, in conjunction with IEEE IRC 2020, entitled as follows,

#CHARM-01 Multi-label UAV sound classification using Stacked Bidirectional LSTM

has been accepted by the CHARMS 2020 Technical Committee after review for technical merits. As part of the publication requirements, it is a mandatory requirement that you attend the Conference to present the paper and discuss your work. Please make the necessary travel arrangements and visa applications as early as possible to be able to present your paper and lead the subsequent technical discussions.

You and your co-authors are invited to attend the conference for presenting the paper at the IEEE IRC 2020 held on March 9-11, 2020 at the Splendor Hotel Taichung, Taiwan.

We are looking forward to seeing you at the conference.

Sincerely,

A handwritten signature in black ink that reads 'Chun-Ming Chang'.

Chun-Ming Chang
On behalf of IEEE IRC 2020

Note – Compiled according to the source [15]

APPENDIX M

Выписка 2 из Протокола № 5 заседания Национального академического совета по приоритетному направлению "Национальная безопасность и оборона" от 11-12 августа 2022 года																		
Дата проведения заседания :										11-12 августа 2022 года								
Дата принятия решения :										12 августа 2022 года								
Председательствовал :										Бердибеков Айдар Токтамысович								
Характер вопроса :										Рассмотрение заявок на конкурс на грантовое финансирование молодых ученых по проекту "Жас Галым" на 2022-2024 годы КН МОН РК								
№	ИРН объекта	Наименование	Заявитель	Научный руководитель	Балл ГНГЭ	Балл ИИС	Дополнительный балл ИИС (баллы за софинансирование)	Общий средний балл	Запланированная сумма на 2022 год	Запланированная сумма на 2023 год	Запланированная сумма на 2024 год	Общая запланированная сумма на 2022-2024 годы	Одобренная сумма на 2022 год	Одобренная сумма на 2023 год	Одобренная сумма на 2024 год	Общая одобренная сумма 2022-2024 годы	Решение Совета	Обоснование
1	AP14971031	Исследование и внедрение биомиметической системы обнаружения беспилотных летательных аппаратов в режиме реального времени.	Некоммерческое акционерное общество "Казакский Национальный Исследовательский технологический университет имени К.И. Сатпаева"	Себдалиева Улжатаг Омуртаевна	27,66	10,93	0	38,59	2995315	7993743	7972269	18961327	2995315	7993743	7972269	18961327	Одобрено	Согласно п.39, п.40 постановления Правительства Республики Казахстан от 16 мая 2011 года №519 «О национальных научных советах», Советом было принято решение одобрить проект.
2	AP14972866	Терроризм и экстремизм в религиозной форме: механизмы реабилитации и дерадикализации в Казахстане и зарубежом.	Некоммерческое акционерное общество "Евразийский Национальный университет имени Л.Н. Гумилева"	Темірбайев Талғат Түмбөлөевич	24,66	11,14	0	35,8	2480070	7658073	7769061	17907804	2480070	7658073	7769061	17907804	Одобрено	Согласно п.39, п.40 постановления Правительства Республики Казахстан от 16 мая 2011 года №519 «О национальных научных советах», Советом было принято решение одобрить проект.
3	AP14971907	Разработка надежной системы обнаружения подзорных БПЛА на тактовой высоте с использованием ЭКВ и акустических сигнатур	Некоммерческое акционерное общество "Казакский Национальный Исследовательский технологический университет имени К.И. Сатпаева"	Утебаева Дана Жолдыбайқызы	23,66	12	0	35,66	2977847,92	7995738,31	7993325	18966911,23	2977847,92	7995738,31	7993325	18966911,23	Одобрено	Согласно п.39, п.40 постановления Правительства Республики Казахстан от 16 мая 2011 года №519 «О национальных научных советах», Советом было принято решение одобрить проект.
4	AP14971555	Проектирование и внедрение системы обеспечения безопасности в режиме реального времени в закрытых помещениях с применением методов машинного обучения	Республиканское государственное предприятие на праве хозяйственного ведения "Институт механики и машиностроения имени академика У.А. Дюлдасбекова"	Дюбаев Жалғол Махсұтұлы	23,33	11,73	0	35,06	2980890	7995322	7966966	18943178	2980890	7995322	7966966	18943178	Одобрено	Согласно п.39, п.40 постановления Правительства Республики Казахстан от 16 мая 2011 года №519 «О национальных научных советах», Советом было принято решение одобрить проект.
5	AP14972687	Независимая антикоррупционная экспертиза нормативных правовых актов как средство обеспечения деятельности правоохранительных органов и защиты национальной безопасности	Учреждение образования "Алматы Менеджмент университет"	Арпын Айжан Арпынқызы	23	11,39	0	34,39	2850630	7342067	7742812	17935509	2850630	7342067	7742812	17935509	Одобрено	Согласно п.39, п.40 постановления Правительства Республики Казахстан от 16 мая 2011 года №519 «О национальных научных советах», Советом было принято решение одобрить проект.

Председателя Совета



Бердибеков Айдар Токтамысович

Figure M.1 – Minute on the acceptance of a scientific project by the "Zhas Galym 2022-2024"