AFEKA The Academic College of Engineering in Tel Aviv

Department of Electrical Engineering

Project Name:

Acoustic Drone Detector

Engineering Report

Student #1

Student Name: Bar Gvili

ID Number: 207299868

Student #2

Student Name: Saar Green

ID Number: 314622341

Student #3

Student Name: Or Meir Cohen

ID Number: 312329113

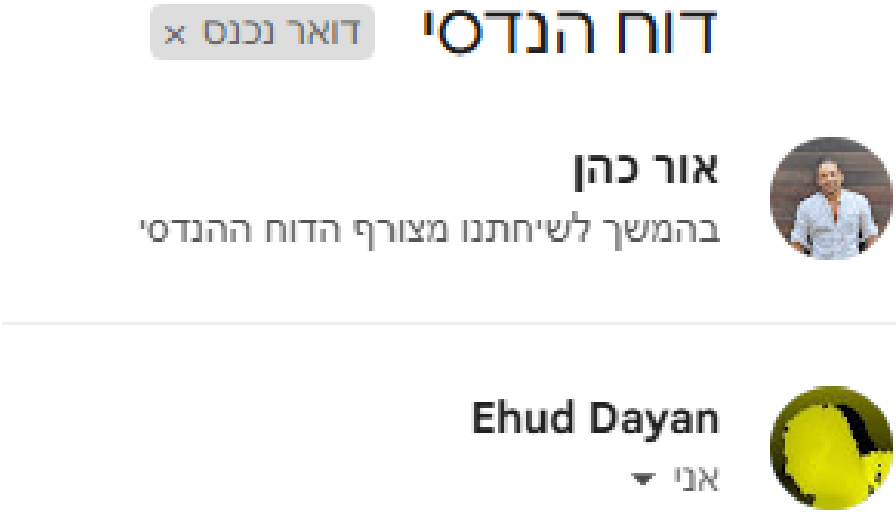
Supervisor's Name: Ehud Dayan

Initiator: Ehud Dayan

Approved By: Ehud Dayan

Submission Date: 16/01/2025

2. approval


דוח הנדסי
דואר נכנס ×
אור כהן
בהמשך לשיחתנו מצורף הדוח ההנדסי
Ehud Dayan
אני
שלום . אני מאשר הגשה
בהצלחה
אודי

3. Table Of Contents

Contents

4. Abstract

The widespread use of unmanned aerial vehicles (UAVs), or drones, has introduced significant security and privacy concerns. Their rapid adoption on the battlefield, driven by advancing technologies that make them increasingly accessible and affordable, presents growing challenges. Concurrently, the increasing use of autonomous drones, which operate without relying on signal transmission or reception, has emerged as a significant threat, as they evade detection by traditional systems. Technologies such as radar and RF-based systems, which do not rely on auditory or visual detection, struggle to identify small, stealthy, or autonomous drones. Recent incidents have highlighted these limitations, with drones successfully bypassing advanced detection systems like **Drone Dome**.

Auditory or visual detection systems offer notable advantages due to their reliance on the physical attributes of drones, such as sound or appearance, making them effective where other technologies fail. For instance, the **Third Eye** visual detection system demonstrates strong capabilities but is limited in poor visibility conditions or in environments with no line of sight, such as urban areas or forests.

The **G2** system is a state-of-the-art acoustic detection solution, utilizing an array of 128 microphones to achieve precise drone detection and tracking. However, its high cost poses a barrier to widespread implementation. The goal of this project is to develop a more affordable acoustic system that provides alerts about the presence of drones in the area and indicates their general direction, rather than precise tracking. This approach can significantly reduce costs, making the system more accessible for security applications and enhancing its practicality for defense purposes.

To meet these objectives, the system must fulfill several key requirements: detecting drones at a range of 25 meters, providing a response time of less than 5 seconds, and indicating the drone's direction within a margin of error of up to 120 degrees. Additionally, the system must achieve a detection accuracy of at least 80%. Success will be measured by the system's ability to reliably detect drones within the specified range and time, while accurately pointing to their general direction with the defined level of precision. Achieving these goals requires a thorough evaluation of various approaches.

Machine Learning (ML) relies on features manually selected by experts, which increases the risk of incorrect feature selection and limits the system's ability to provide a complete representation of the data. This approach contrasts with Deep Learning (DL), which autonomously learns directly from raw data, deriving insights without pre-defined features. DL's ability to extract meaningful patterns makes it particularly effective in complex and dynamic scenarios.

For feature analysis, time-frequency representations were chosen as they integrate information from both time and frequency dimensions, providing a holistic view of the signal. Other approaches, such as time-only or frequency-only features, were found less suitable, as they risk losing critical information needed for optimal drone detection. To process time-frequency features, image-based models were utilized, treating spectrograms as images to leverage existing architectures optimized for handling complex image structures. As detailed in [96], CNNs demonstrated high detection accuracy but suffered

from longer response times. RNNs excelled in identifying temporal patterns with faster processing speeds, but their accuracy was lower. CRNNs, combining the advantages of CNNs and RNNs, achieved the best balance between accuracy and response time, making them the preferred choice for this project.

The choice of hardware also played a critical role in the system's development. Raspberry Pi 4 was selected as an initial cost-effective solution, meeting the project's requirements within budget constraints. If it fails to support real-time processing, Jetson Nano, offering superior processing capabilities at a higher cost, will serve as an alternative. Regarding the acoustic capture, regular microphone arrays provide a form of "digital zoom," amplifying signals effectively when their intensity surpasses the sensitivity of individual microphones in the array. In contrast, parabolic microphones employ "mechanical zoom," physically focusing and amplifying signals through a geometric structure, enabling effective detection over extended distances. This unique capability made parabolic microphones a viable choice for achieving the project goals.

The parabolic dish has unique traits: its cutoff frequency depends on its size, and below that threshold, its reception efficiency drops considerably. Additionally, each frequency is amplified differently, and the dish only accepts signals within a defined angular range (the beamwidth). Before constructing a dish that fits our needs, we must first assess the frequency behavior of drones and examine how sound attenuates in open space.

Sound energy weakens primarily through spherical spreading, which impacts all frequencies equally, and absorption, which is frequency-dependent. By exploring drone spectrograms and calculating sound attenuation across our intended detection range, we see that high frequencies are sufficient to identify drone patterns. Even a quiet drone generates about 65–70 dB, and to maintain a safety margin, we design the system to identify sounds around 60dB. Based on these findings, we require a parabolic dish that provides at least 10 dB of gain and limits its beamwidth to no more than 120 degrees. Because sound waves (unlike radio waves) do not demand specialized materials or an absolutely perfect surface, 3D printing becomes a practical manufacturing solution. This dish will allow us to detect drones from a greater distance, reduce noise through directional focusing, and capture the essential audio signals that feed into our model.

We then shift to time-frequency domain features, which involve segmenting the signal into brief time windows, transforming each into the frequency domain, and combining them into a single spectrogram. Determining the minimum window size starts with understanding drone frequency ranges; from that analysis, we chose 100 milliseconds to cover both typical drone frequencies and an additional safety buffer. Among the methods tested (STFT, MFCC, Mel Spectrogram), we began with Mel spectrograms and MFCCs, which mirror how humans perceive sound and help us visually verify what we hear. For instance, if a person can identify the drone, the model should recognize it too. If further details are needed, we can switch to regular spectrograms (STFT), but they're heavier computationally.

Finally, Mel spectrograms are treated as images and processed using a CRNN (Convolutional Recurrent Neural Network). We'll start by training on public data and then incorporate fresh recordings made with our parabolic dish to refine the model's performance. This will enhance detection accuracy by tailoring the data to our custom dish's directional characteristics.

The Result.

A prototype parabolic dish measuring 0.33[m] in diameter was 3D-printed and tested, offering a maximum gain of 25 dB and an average gain of 15 dB. Although not yet integrated into the final detection model, test results demonstrate that a drone can be heard clearly at a distance of 100 meters, meeting the project's range requirements, the dish provides a maximum beamwidth of 60 degrees, which outperforms the initial target (up to 120 degrees). By deploying multiple dishes, it becomes possible to pinpoint the drone's direction with an error margin significantly lower than 120 degrees.

In parallel, a drone detection system was developed using one-second audio recordings. Each one-second clip is transformed into a Mel spectrogram, which is then fed to a CRNN model. The spectrogram conversion and classification process takes no more than 100 milliseconds, so the total system response time—recording plus processing—is about 1.1 seconds. This is well within the project's 5-second maximum detection requirement.

The model was evaluated on multiple online datasets as well as a smaller set of in-house recordings. Its current accuracy stands at 95%, surpassing the original 80% success rate goal. Below are the detailed classification results:

	precision	recall	f1-score	support
0	0.97	0.94	0.96	781
1	0.93	0.96	0.95	612

Accuracy: 0.95 Table 4.1

Details:

- **Precision:** Measures the proportion of true positives among all predicted positives. For drones, the precision is 93%, while for non-drones, it is 97%.
- **Recall:** Measures the proportion of true positives among all actual positives. The recall for drones is 96%, indicating a low rate of false negatives.
- **F1-Score:** The harmonic mean of precision and recall, providing a balanced measure. Both classes achieve high F1-scores of approximately 0.95–0.96.
- **support:** refers to the number of true instances for each class.
- **Accuracy:** The overall model accuracy is 95%, which is significantly higher than the 80% detection rate required by the project objectives.

Moving forward, the project will focus on several key objectives. First, the dataset will be expanded by collecting a broader range of recordings and exploring additional acoustic features. Another avenue of research involves generating synthetic audio samples using a GAN model to enrich the training data and further improve model performance. Efforts will also be made to reduce the sampling time, aiming to optimize real-time detection capabilities. In parallel, the dish's performance will be evaluated under various conditions. Finally, the prototype will be deployed on platforms like the Jetson Nano or Raspberry Pi, creating a lightweight, easily deployable solution for field operations

5. Introduction

5.1 Project Essence

Drone attacks are a growing problem worldwide, Israel has also been affected from this and since October 7th, there has been a significant increase in attacks by various drones and UAVs, launched by terrorist organizations such as Hamas and Hezbollah. The challenge of detecting and identifying these tools is considerable, given their small size, low heat signature, and the fact that many autonomous drones do not emit radio waves.

The suicide drone has become an integral weapon in modern warfare, posing a significant threat to infantry soldiers on the battlefield. The increasing complexity and variety of these drones underscore the urgent need for effective detection systems to enhance security and protect against these evolving threats.

5.2 target audience

The primary audience for this project includes security and defense organizations, both governmental and private, that are focused on developing advanced solutions for detecting and mitigating drone threats. This includes military and defense agencies, which require effective detection systems to protect soldiers, infrastructure, and operational integrity in combat zones. Law enforcement and border security agencies also need tools to identify and respond to drone threats, particularly in high-risk areas such as borders and sensitive facilities.

5.3 Project Goals, Objectives, and measurements

Goals

- Detection of UAVs or drones in the environment using acoustic methods.
- Providing information on the direction of the drone.
- Gaining a deep understanding of signal processing and machine learning.

Objectives

- Detection of a drone at a distance of 25 meters and a UAV at a distance of 200 meters.
- Implementation of an analog circuit for signal processing.
- Creation of a custom artificial intelligence model specifically designed for the task.
- Development of a portable wireless system.

measurements

- Correctly identify the drone in 80% of the cases.
- Ensure the response time is no more than 5 seconds.
- Point to the drone's direction within 120 degrees.

5.4 Analysis of the solution space

1. **Detection Using Radar method**- Radar-based drone detection utilizes electromagnetic waves to detect and locate objects. It operates by sending out short electromagnetic waves and analyzing the signals reflected back from the target. This system includes a transmitter, receiving antennas, a radar receiver, and a processor that determines object attributes such as distance, velocity, azimuth, and elevation. One of the challenges with radar-based drone detection is distinguishing between small objects with low radar cross sections (RCS), like drones, and other similarly small objects. Another challenge is detecting drones that fly at low altitudes, as they may move below the radar's scanning sector. [1]

2. **Detection Using RF method-** RF-based systems detect drones by monitoring the radio frequency signals that drones emit during their operation, such as those used for communication between the drone and its controller or GPS systems. These systems continuously scan the airwaves, listening for specific RF signatures that indicate the presence of a drone. Once detected, the system can track the drone's position and movement. Drones usually communicate with their controllers using RF signals, often in the 2.4 GHz ISM frequency band. Since the specific frequency of a drone is typically unknown, an RF scanner passively listens to the signals exchanged between the drone and its controller. This approach is particularly effective in detecting UAVs in restricted areas by analyzing the RF signals for control commands and data transmission between the UAV and its ground station. A key challenge with RF-based detection is that it relies on the drone being actively transmitting RF signals. If the drone is not transmitting, it may go undetected. Moreover, the system can be susceptible to interference from other RF sources, and its detection range may be limited compared to other methods like radar. [1]

3. **Detection Using Vision-based method-** Vision-based drone detection systems use cameras to visually monitor the sky, looking for drones through image processing and pattern recognition techniques. These systems can detect and track drones by analyzing the movement, shape, and size of objects captured in the video feed. They are particularly effective in clear conditions and can provide real-time visual confirmation of a drone's presence. The process involves three steps: reflecting energy from the object, focusing it with an optical system, and measuring it with a camera sensor. The main challenge with vision-based detection is that it requires good visibility, which can be compromised by poor weather conditions such as fog, rain, or darkness. Additionally, distinguishing between drones and other small flying objects can be difficult, especially when the objects are far away or moving quickly.

4. **Detection Using Acoustic Method-** Acoustic-based drone detection identifies drones by analyzing the distinctive sounds they produce, such as the noise from their engines and propeller blades. These systems use highly sensitive microphones or microphone arrays to capture and analyze the acoustic signatures of drones, focusing on characteristics like frequency, amplitude, and modulation. By examining these audio signals and comparing them to a database of known drone sounds, the system can determine the presence of a drone and, in some cases, assess its type and capabilities. A primary challenge with acoustic detection is that it can be affected by environmental noise and may have limited effectiveness in noisy or windy conditions. Additionally, it requires the drone to be generating sound, meaning silent or very quiet drones could evade detection.

Drone Detection Technologies Comparison

Attribute	**Radar**	**Acoustic**	**Electro-Optical (EO)**	**RF Detection**	**Lidar**
Range (average)	1-5 km	300m-1 km	500m-2 km	2-10 km	100m-1 km
Advantages	Works in all weather, long-range	Effective for small drones, low false positives	High precision, visual identification	Detects communication, long-range	High accuracy in 3D, small object detection
Disadvantages	Struggles with small drones, false positives	Short range, affected by noise	Limited in poor visibility	Ineffective for autonomous drones	Shorter range, expensive
Autonomous Drones Detection	Limited (no RF signals to track)	Effective	Limited (visual-only)	Ineffective	Limited (line of sight issues)
Power Consumption	High	Low to moderate	Moderate to high	Low to moderate	High
Mobility	Typically stationary but can be mobile	Highly mobile, lightweight	Can be portable	Portable and lightweight	Mobile but generally larger systems
Size	Medium to large	Small arrays	Compact to medium	Small to medium	Medium to large
Cost (estimate)	$100K-$500K+	$50K-$200K	$100K-$300K	$50K-$300K	$200K-$1M+

Table 5.4.1

Drone Detection Technologies - Scores

Attribute	Radar	Acoustic	Electro-Optical (EO)	RF Detection	Lidar
Range	8	4	5	9	6
Mobility	5	9	7	8	5
Size	2	9	7	7	5
Small Object Detection	4	8	7	3	8
Autonomous Drone Detection	5	8	6	0	5
Price	4	7	5	6	2
Power Consumption	3	7	5	7	3
No Line of Sight	6	8	0	7	0
Weather	9	6	3	8	4
Final Score	6.1	7.6	5.5	6.1	4.7

Table 5.4.2

5.5 The Use of AI:

Within the realm of technological advancements, machine learning has emerged as a prominent tool across various domains, particularly in the realm of object detection and classification [2]. Machine learning algorithms possess the capacity to autonomously acquire knowledge and identify patterns without requiring constant human intervention. Moreover, they are capable of capturing information that may evade human perception, including radio frequencies and audio signals within specific ranges. Extensive research has been undertaken to explore visual, radar, radio-frequency, and audio-based methodologies, each yielding promising outcomes. Nonetheless, it is vital to recognize that each approach possesses its own set of strengths and limitations.

Audio processing technology plays a ubiquitous role in our daily lives, as exemplified by the prevalence of popular products like Apple's Siri, Amazon's Alexa, and Google Home Mini Dot, which leverage audio processing and artificial intelligence (AI). AI serves as the underlying mechanism enabling computers and smartphones to comprehend human speech, thus facilitating effective interaction between humans and machines [3]. At the core of audio-based intelligent systems lies the ability to listen to and interact with the environment, continuously learning and enhancing their responses. Such intelligent systems find applications in various domains, including smartphone applications that engage users through natural language interfaces, or computer software capable of identifying bird species based on their vocalizations in a backyard setting. Figure 5.5.1 presents an overview of the fundamental processing structure of an audio-based machine learning (ML) system, encompassing key steps such as audio data pre-processing, windowing, feature selection and extraction, and classification[4]. Initially, the system receives raw audio data samples as input, which then undergoes a pre-processing step to address concerns such as noise reduction, cancellation, or normalization. Subsequently, a windowing function is applied to facilitate analysis and comprehensive examination of the entire audio sample. The choice of windowing methods may vary depending on the specific characteristics of the audio data. Feature extraction and selection represent the subsequent phase, where relevant features are identified to serve as input for training the ML model. Finally, the trained classifier leverages these features to make accurate predictions.

Figure 5.5.1 ML structure for audio

Artificial Intelligence (AI) refers to the intelligence exhibited by machines, such as computers, enabling them to mimic human behavior. Various techniques are employed in AI, including machine learning, computer vision, natural language processing (NLP), and robotics [5]. One prominent approach to achieving AI is through Machine Learning (ML), which involves training computer systems to learn from experience and enhance their performance over time. Neural Networks (NN) form a significant subset of ML, simulating the functioning of the human brain by processing input data through interconnected neurons or nodes. On the other hand, Deep Learning (DL) is a specialized branch of NN that necessitates the inclusion of multiple layers within the network's structure. The focus of this paper revolves around the exploration of deep learning architectures and their applications in the field of audio classification.

5.6 Engineering Challenges

- **Identifying Propeller Signatures Across Frequencies**
  It is challenging to rely on data from specific frequencies alone to accurately identify drone propeller signatures. The system needs to effectively distinguish these signatures across a range of frequencies.

- **Developing Highly Sensitive Microphones**
  Designing and implementing microphones that can detect very low sound signals is crucial. These microphones must be capable of capturing the specific acoustic signatures within the range required for effective drone detection.

- **Massive Noise Filtering**
  Filtering out background noise is essential to ensure the accuracy of the detection system. This involves developing sophisticated noise reduction techniques to isolate the drone's sound from environmental sounds.

- **Building a Comprehensive Dataset**
  Creating a robust dataset for training the detection model is necessary. This includes collecting and analyzing various acoustic features to improve the model's ability to extract relevant information from the audio signals.

- **Real-Time Processing**
  The system must process audio signals quickly enough to provide real-time detection. This requires high-speed data processing capabilities to ensure timely and accurate identification of drones.

5.7 Division of Work

Bar: Responsible for all hardware-related tasks, including understanding microphone characteristics and their reception spectrums, selecting the necessary hardware components, and developing a custom microphone array for the project. In addition, fallback solutions for the microphone array will be prepared in case the primary design encounters any issues.

Saar: Focuses on feature extraction, dataset creation, and signal processing. This involves conducting recordings, gathering and organizing relevant data, and applying DSP (Digital Signal Processing) techniques to process the signals.

Or: Responsible for the overall system architecture and integration, including the development of machine learning models and the GUI. This role involves defining system requirements and ensuring seamless integration among hardware, signal processing, and software components. A key responsibility is to ensure all system elements function together cohesively.

6. Literature Review

6.1 Scientific literature

6.1.1 Machine and Deep Learning

background

Machine Learning (ML) algorithms can be categorized into three groups: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning refers to a specific type of machine learning that utilizes labeled datasets to train models for solving classification or prediction problems. The model's weights are adjusted iteratively until it fits the input data accurately. Supervised learning finds applications in various fields and domains. For instance, in the context of a mobile phone, supervised learning can enable the identification of the music being played on the radio by analyzing a few seconds of audio.

On the other hand, unsupervised learning involves the utilization of unlabeled datasets, typically for the purposes of analysis and clustering. These algorithms are capable of discovering patterns and categorizing data without requiring human intervention. Unsupervised learning algorithms are particularly valuable for uncovering hidden differences and similarities within the data. This capability enables the solution of real-world problems such as pattern recognition and anomaly detection.

Reinforcement learning, another important category of ML, places a strong emphasis on the concepts of reward and action. It involves rewarding desired behaviors and/or penalizing errors. Reinforcement learning has been extensively studied and applied across various domains, including control theory, multi-agent systems, and statistics. Notably, it has also found wide-ranging applications in autonomous driving scenarios.

Deep Learning Models in Audio-based Applications

Deep neural networks (DNNs) have gained popularity in audio-based research and applications due to their remarkable performance and ability to handle large datasets. In the subsequent paragraphs, we will delve into several significant DNN architectures, including Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Generative Models [6]. Figure 6.1.1 illustrates the diverse structures of deep learning models.

CNNs are commonly used in audio processing tasks, particularly for tasks like speech recognition and audio classification. These networks excel at capturing local dependencies through convolutional layers, enabling them to extract meaningful features from audio signals effectively [6]. RNNs, on the other hand, are designed to handle sequential data, making them suitable for tasks involving temporal dependencies. In audio applications, RNNs are extensively used for tasks such as speech synthesis and music generation. The recurrent connections in RNNs enable them to retain and utilize information from previous time steps. Generative models, such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs), have shown great potential in audio synthesis and audio generation tasks. These models learn the underlying data distribution and generate new audio samples that exhibit similar characteristics to the training data.

The versatility and adaptability of these DNN architectures have greatly advanced audiobased research and applications, revolutionizing areas such as speech processing, music analysis, and sound synthesis.

Convolutional neural networks (CNNs) are feed-forward networks consisting of multiple layers of neurons/nodes. They were specifically designed to process data with grid-like topologies, such as images [7]. CNNs, in conjunction with computer vision techniques, have consistently achieved state-of-the-art results in various image processing tasks, including classification, detection, segmentation, and more [8].

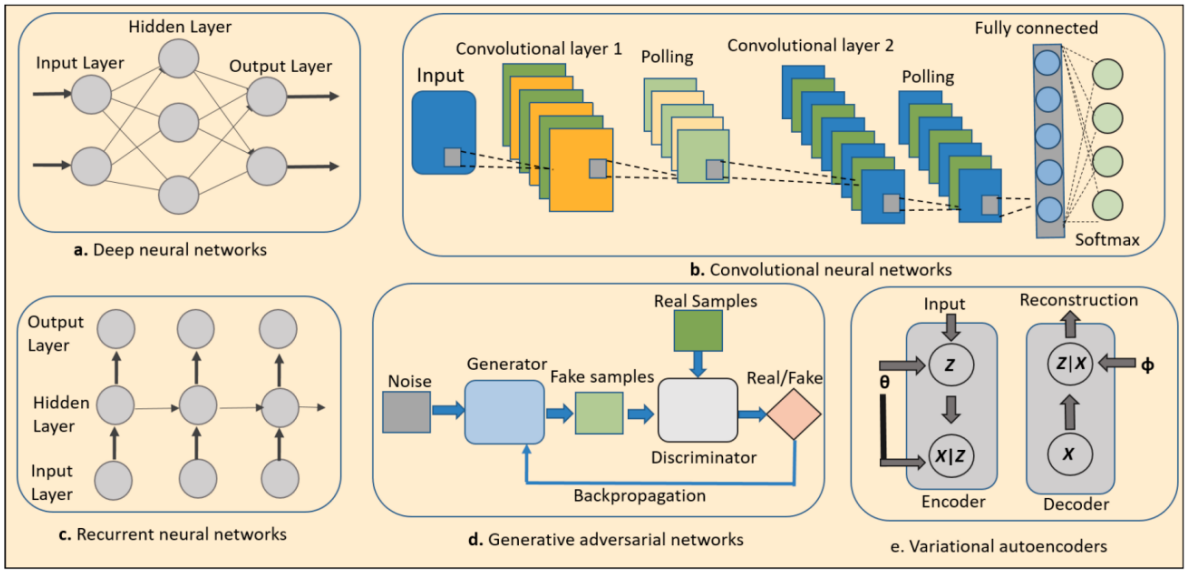


Figure 6.1.1 Different structures of deep learning models[9]

Unlike traditional deep neural networks (DNNs), CNNs exhibit memory and parameter efficiency due to two key reasons: local receptive fields and shared weights. CNNs typically comprise multiple convolutional layers followed by one or more dense layers. However, fullyconvolutional networks (FCNs) exclude the dense layers, resulting in even fewer parameters. FCNs, along with their extensions, enable domain adaptation and enhance the robustness of the network [10].

CNN models have found applications in various audio processing tasks, such as automatic speech recognition (ASR) [11], music genre classification [12], and speech enhancement [13]. Nevertheless, when it comes to processing raw audio waveforms with high sample rates, the limited receptive fields of CNNs can present challenges [9]. Dilated convolution layers have emerged as a solution to address this issue. They expand the receptive field by inserting zero values between the filter coefficients, allowing for a larger effective receptive field [14].

In summary, CNNs have demonstrated their effectiveness in image processing tasks, leveraging their ability to exploit local structures in grid-like data. In audio processing, CNN models have been successfully employed in various applications, but the limited receptive fields of CNNs can pose challenges in handling high sample rate audio. Dilated convolution layers over a solution by extending the receptive field to effectively process raw audio waveforms.

Recurrent neural networks (RNNs) have a different approach to processing sequential data [15]. The use of recurrent connections between layers enables parameters to be shared recurrently. This unique approach makes them efficient and powerful in understanding and learning temporal data structures from the sequential data input, such as audio and video input [9]. Compared to the traditional Hidden Markov model (HMM) models, RNNs have produced better results in many audio and speech processing applications [16]. Because of these characteristics, two of the most popular RNN structures, Long-Short Term Memory (LSTM) [17] and Gated Recurrent Unit (GRU) network, significantly improved the audio and speech processing applications and were used to build state-of-the-arts audio-based systems [18]. In recent years, Time-Frequency LSTMs [19] and Frequency-LSTMs [20] were created based on previous RNN models with information in the frequency domain. To take advantage of both neural networks, Convolutional Recurrent Neural networks (CRNN) were created by combining CNNs and RNNs with convolutional layers followed by recurrent layers [16]. CRNN has been used in music classification[21], ASR [22], Speech Emotion Recognition (SER) [23], and more.

Recent Research on DL-based Audio Application

Automatic speech recognition (ASR) is to use algorithms to convert a speech, usually in the form of audio, into text. Contemporary ASR systems have achieved significant results because of the use of DL models, with extensive supervised training and a large number of labeled training data. To explore more efficient solutions, RL-based models were also used in ASR, for their capability of learning from action. RL-based ASR systems can generate positive or negative rewards instead of manually preparing these by human [24][25][26]. [24] proposed a policy gradient-based RL system for ASR. They provided another angle for existing training and modification methods. The proposed system achieved better recognition performance and lower word error rate (WER) than unsupervised methods.

Audio-based intelligent systems are extremely sensitive to environmental noise, and the system's accuracy decreases when the noise level goes up [27]. Audio enhancement is one of the possible solutions for noise interference. Audio enhancement systems are supposed to filter out the noise and generate an enhanced audio signal. DL-based models have achieved significant performance in speech enhancement, compared to the traditional methods [28]. [28] proposed an RL-based speech enhancement system to advance adaptivity. They designed the noise-suppression module as a black box, that does not need to understand the algorithm but provides simple feedback from the output. They achieved better performance with the LSTM-based agent. [29] proposed a DRL-based approach for hearing aid application. The proposed system can tune the compression from noisy speech according to the individual's preference. Human hearing is non-linear. The system adopted DRL's reward and punishment rule and the DRL model receives preferences from the hearing aid user. Results indicated that the proposed system improved the hearing experience and the user was satisfied with the hearing outcome.

DL-based systems have also been used in generating more data content, such as images, music, and text. DL models were first used in music generation because of their ability to learn and compose (generate) any genre of music from existing music database [30][31]. The DRL-based intelligent system can achieve more and provide more ways of learning directly from music theory to compose music with structures that sound more like real ones [30]. [31] proposed an LSTM-based model that can compose polyphonic music based on music theory with better quality. [30] proposed a system of deep Q-learning structure with a reward function that learns from the probabilistic outputs of an RNN and basic rules of music theory. The results indicated that the proposed model can learn to compose and keep the valuable information of data from supervised training.

Sound classification is another application for DL-based audio systems. It can be used in specific tasks, such as bird sound classification [32], environmental sound classification [33], music classification [32], and more. [32] proposed a DL-based bird sound classification system. They utilized CNN for learning generalized features and dimension reduction, with a conventional fully connected layer for classification. The proposed DL approach outperforms the other methods, including acoustic and vision-based systems. But they achieved the best result from combining all visual, acoustic, and DL learning. [33] proposed a deep CNN structure for environmental sound classification. Furthermore, they used data augmentation techniques to compensate for the lack of publicly available datasets and investigate the performance of different augmentation techniques with the proposed deep CNN structure. With the data augmentation and proposed CNN network, they achieved state-of-the-art classification results. [34] evaluated DL-based CNN models and feature engineering-based models for music genre classification. CNN structures included VGG-16 CNN with transfer learning, VGG-16 CNN with fine-tuning, and fully-connected NN. Feature engineering-based models were logistic regression (LR), random forest (RF), support vector machines (SVM), and extreme gradient boosting (XGB). They also built an ensemble classifier with CNN and XGB. Results indicated that DL-based models had better classification accuracy, and the best result was achieved by the ensemble classifier.

6.1.2 Audio Data Pre-Processing and Augmentation

Data Pre-Processing

Audio data needs to be pre-processed before feeding into the ML models. In gradient descent-based algorithms, feature standardization is commonly used to accelerate the process of convergence [35][6]. Feature distribution is changed from feature standardization with zero mean and unit variance. A large dynamic range usually appears in environmental sound data. A commonly used solution is logarithmic scaling applied to spectrogram-based features. Pre-processing methods for low-level audio signals include low-pass filtering and speech dereverberation [36].

In many audio-based applications, such as automatic speech recognition (ASR) and acoustic event detection (AED), background noises sometimes overshadow the foreground sound events. [37] proposed to use per-channel energy normalization (PCEN) to enhance foreground sound events and reduce background noise in environmental audio data. The proposed system adjusted the PCEN parameters with the temporal features of the noise to reduce the noise level, while the foreground sound signal is enhanced. [38] proposed to use two edge detection methods from image processing to enhance the edge-like structures in spectrograms. Those two methods were based on the difference between Sobel filtering and Gaussians (DoG). The Median filter is used to remove the drift of the Mel spectrogram.

Commonly-used pre-processing methods for ASC applications are filtering methods. [39] proposed an ASC system that included a nearest neighbor filter based on the repeating pattern extraction technique (REPET) to filter out repetitions appearing intermittently or randomly. The most similar spectrogram frames were replaced by their median. On the other hand, this filter can be used to highlight repetitive sound events in AED, such as horns and sirens. Another commonly used filtering method is harmonic-percussive source separation (HPSS). HPSS splits the spectrogram into horizontal and vertical modules that provide additional features for ASC [40]. All the above pre-processing approaches were relatively new, compared to the well-established and most-used logarithmic magnitude scaling among the state-of-the-art ASC algorithms [35].

Data Augmentation

A large amount of training data is essential for deep learning models to learn. In recent years, the datasets for audio classification are increasing, but still not as much as the image datasets [35], such as ImageNet [41]. As far as today, the largest audio dataset is AudioSet, which includes 632 audio classes and a collection of almost 2 million labeled 10s excerpts from YouTube video [42]. But still, there is a need for more publicly available audio datasets. Many researchers have been trying to compensate for this issue with data augmentation techniques. There are mainly two different approaches: to generate new data based on existing ones and to generate synthetic data from scratch.

The first kind of data augmentation is to generate new training data based on existing ones with added signal transformations. Commonly used audio signal transformation methods are pitch shifting, time stretching, and adding noise [43]. [44] proposed to use spectral rolling and mix-up to augment the audio data. The former technique randomly shifts the spectrogram features over the time dimension, and the latter one works linearly by combining features from the data and their targets with a given mixing ratio [45] proposed a simple data augmentation method called SpecAugment, which is applied directly to the feature (log Mel spectrogram) of the audio data. The augmentation policy included warping the features, frequency masking, and time masking. [46] used various data augmentation techniques on both the time and frequency domain. For the time domain data augmentation, there are mosaicking random segments, time stretching, time interval dropout, and more. And for the frequency domain, they used frequency shifting/stretching, piece-wise time, resizing filters, and color filters.

The other kind of data augmentation technique is to generate synthetic data from scratch. The most popular approach for this kind is to use generative adversarial networks (GAN) [47]. An adversarial training strategy was used to train synthesizing models by mimicking the existing audio data. Most data synthesis techniques are applied to the audio signal [35].

6.1.3 Feature Extraction Methods for Audio

background

Feature extraction is the procedure of articulating the most representative and refined characteristics of audio data. Proper features can present the audio data in a considerably compact manner. The evolution of audio features can be divided into four categories: time domain features, frequency domain features, joint time-frequency domain, and deep features. The first kind of features extracted from audio data is time domain features, which are also the simplest kinds, the time domain features were discovered around the 1950s. Time domain features were widely used in audio/acoustic analysis and audio-based classification since then. Between the 1950s to 1960s, frequency domain features, like formant and pitch, were discovered and adopted in different audio-based applications since then. From the late 1960s, the joint time-frequency features were discovered and used in various audio-based systems and applications. Examples include the short-time Fourier transform (STFT) and the wavelet transform. Because of the development of artificial intelligence (AI) and deep learning (DL), deep features of audio data are widely studied and used in many audio-based applications, such as acoustic scene classification, audio/video analysis, and speaker recognition since 2010.

Feature Extraction

There are numerous types of features that can be extracted from audio signals for various applications, such as speech recognition, music analysis, etc. In this section, we focus on the features most relevant to the acoustic detection of unmanned aerial vehicles (UAVs). These features capture important aspects of the sound, enabling systems to classify and distinguish between background noise and the unique acoustic signature of drones.

Window Function

The most straightforward way to analyze an audio signal is via its original form [48] [6]. All the audio signals discussed are time series signals, which means signals that develop over time.

After we visualize a signal in the time domain, key characteristics of the signal can be analyzed, which can be used in predicting and comparing with similar signals. However, the real-time audio signals are non-stationary over time, which cannot be analyzed by using time domain analysis. Windowing techniques are required to analyze non-stationary signals, and long non-stationary signals are analyzed in chunks of quasi-stationary signal [48]. Windowing is to apply a window function on a signal. A window function is to apply zero to the area that is outside of the interest time period of an audio signal. The area inside of the interest time period is non-zeros [49]. The outcome of a windowed signal is a subset of the original signal that passed through the window, as the rest of the signal is zero. The most basic type of window is a rectangular window. The problem with using a rectangular window is the sudden change at the edges of the window, which might create distortion when analyzing the signal. The distortion is caused by the Gibbs phenomenon [50]. The more advanced window functions, such as hamming or hanning window, can reduce or avoid the Gibbs phenomenon and smooth the curves of the signal [49]. These window functions are at 0 on the edge of the window but increase gradually to become 1 in the middle of the window.

Time Domain Features

Zero-crossing rate (ZCR)

The zero-crossing rate is the number of times in a given time frame/interval that the amplitude of an audio signal passes through the value of 0 [51]. ZCR can be used to detect voice activities, such as whether a frame of speech is voiced, silent, or unvoiced. The number of ZCR is lower for voiced activity compared to the unvoiced ones. ZCR can also be used to estimate the fundamental frequency (FF) of a frame of speech [52]. Thus, ZCR can provide indirect information about the frequency of the audio signal. ZCR has been used to develop classifier and discriminator [53], music genre classification, voice activity detection [54], and vowel detection and analysis [55].

Amplitude Descriptor (AD)

The amplitude descriptor (AD) is one of the amplitude-based features, which are based on basic analysis of the temporal envelop of the signal [48]. AD distinguishes various types of sound envelopes in the aspect of energy. It separates the signal into low and high amplitude by an adaptive threshold (a level-crossing operation) [56]. AD has mainly been used in environmental sound classification and animal sound classification.

Log Attack Time (LAT)

The log attack time feature is also a type of amplitude-based feature. It is logarithmic with base 10 of the time duration from when the sound becomes perceptually audible to when it reaches its maximum intensity[57]. It has been used in environmental sound classification [58] and music onset detection [59].

Shimmer

Another type of amplitude-based feature is shimmer. It calculates the average absolute difference between the amplitudes of the continuous periods, divided by the average amplitude [60]. It has been used in stress and emotion classification [61], speaker detection and verification [62], and music sound classification [63].

Short Time Energy (STE)

The energy within the signal is constantly changing. Thus, it is not useful to learn from or to predict a value. Because of this, the energy from a frame is calculated and called the short-time energy. STE is a type of energy-based feature. STE describes the envelope of a signal [64]. The number of STE is high for the voiced segment and low for the unvoiced segment. STE has been used in environmental sound detection [65], audio-based surveillance systems [66], music onset detection [67], and vowel detection and analysis [55].

Volume

Volume is another type of energy-based feature. The volume of a sound is one of the most straightforward features of the human auditory system. It has been used in acoustic scene classification [68], speech and music classification [69], and speech segmentation.

Temporal Centroid (TC)

The temporal Centroid indicates where the center of mass of the spectrum is located. It has been used in environmental sound classification [70] and acoustic scene classification [71].

Auto-correlation Based Features

The autocorrelation is the correlation of a signal with a delayed replica of itself, as a function of delay [4]. In other words, it indicates the similarity between the signal and its delayed version. Auto correlation-based features have been used in acoustic scene classification [72], and music tempo and beats estimation [73].

Rhythm-based Features

The rhythm defined as a strong, regular, repeated pattern of a sound over time [74]. A rhythm can be found in musical compositions, poetry, and environmental sound, like in bird songs. Rhythm-based features include articulation rate, speech duration, pause ratio, total vowel duration, beat histogram, and more [4]. Rhythm-based features have been used in music genre classification [75], music instrument classification, speech/music discrimination [69], and analysis of pathological speech [76].

Frequency Domain Features

The time domain features indicate the change of audio signal in terms of time. To analyze the change of a signal in terms of frequency, we convert the time domain signal to a frequency domain signal by using Fourier transform or auto-regression analysis [77]. Frequency domain features are the most important ones in audio signal analysis and processing [4]. Table 6.1.3 shows the selected frequency domain features and their applications.

Peak Frequency

The peak frequency is simply the frequency of maximum energy. It estimates the most dominant frequencies present in the signal and helps to calculate the fundamental frequency of the signal [4]. It has been used in music and speech classification [75] and gender classification [78].

Chroma Based Features

Chroma features are interesting and powerful representations of music audio, where the entire spectrum is divided into 12 parts, representing the 12 semitones, or 12 chromatic tones, of an octave in music notation. It can be calculated from the logarithmic short-time Fourier transform of the sound signal. It is also called a chromatogram. Another chroma-based feature is the chroma energy distribution normalized statistics. This feature is used to identify similarities between different interpretations of a given piece of music. Chroma-based features have been used in music genre classification [79].

Spectral Centroid

The spectral centroid indicates where the spectral centroid is located. It represents the brightness of the sound signal, so it is also called the brightness characteristic of the sound. The calculation of spectral centroid treats the spectrum as a distribution where the values are frequencies and the probability of observing those values is the normalized amplitude. The spectral centroid feature has been used in music classification [80], music mood classification [81].

Time-frequency and Cepstral Features

The cepstrum is calculated by taking the inverse Fourier transform of the logarithm of the signal spectrum. There are complex, power, phase, and real cepstrums. Of all these, the power cepstrum is the most relevant feature to speech and audio signal processing. Cepstral features have been widely used in different audio processing applications and systems. Table 6.1.3 shows the selected cepstral features and their applications.

Short-Time Fourier Transform (STFT)

The Short-Time Fourier Transform (STFT) is a commonly used technique for representing audio signals in the time-frequency domain. It is computed by segmenting the signal into overlapping frames, applying a window function to each frame, and performing the Fourier Transform. The result is a spectrogram where time, frequency, and intensity represent the temporal evolution of the signal's frequency content. The STFT serves as a powerful feature extraction method in audio processing tasks, enabling the transformation of raw audio into a representation suitable for machine learning models.

STFT is widely employed as a primary feature in training models for various audio-related tasks. For instance, it has been used for respiratory abnormality detection in lung sounds, where STFT features were input to a fine-tuned ResNet18 model for classification [82]. Additionally, STFT was compared with other time-frequency representations for environmental sound classification using convolutional neural networks (CNNs), demonstrating its effectiveness in capturing meaningful audio patterns [83].

The flexibility and detail provided by STFT make it a foundational feature extraction method in modern audio processing systems.

Mel Spectrogram Features

The Mel spectrogram is a time-frequency representation of an audio signal that maps frequencies onto the Mel scale, aligning with human auditory perception. To compute a Mel spectrogram, the audio signal undergoes a Short-Time Fourier Transform (STFT), and the resulting frequency components are passed through a Mel filter bank. The energy in each Mel band is computed and logarithmically transformed, creating a spectrogram where the axes correspond to time, Mel frequencies, and log amplitude.

Mel spectrograms are extensively used as primary features in training machine learning models for various audio processing tasks. For instance, Mel spectrograms have been used as input to convolutional neural networks (CNNs) for audio recognition tasks, demonstrating their effectiveness in capturing the time-frequency characteristics of audio signals [84]. In audio classification, Mel spectrograms have been applied to deep learning models to identify and categorize audio data accurately [85]. They have also been widely utilized in speech processing tasks, providing a perceptually relevant representation of audio for various machine learning applications [86].

The effectiveness of Mel spectrograms in highlighting perceptually significant features has established them as a preferred choice for feature extraction in modern audio analysis systems.

Mel Frequency Cepstral Coefficients (MFCCs)

The Mel Scale is a logarithmic transformation of a signal's frequency. The main idea of this transformation is that sounds that are equidistant on the Mel scale are considered to be equidistant from human hearing. The Mel spectrograms are spectrograms that use Mel scale to visualize sound. To obtain MFCCs from an audio sample, first, we need to convert the audio from Hertz to Mel scale. And then we need to take the logarithm of the Mel representation of audio, followed by taking the logarithmic magnitude and applying discrete cosine transformation, following the mentioned steps, a cepstrum created over Mel frequencies and is called MFCCs [87].

MFCC is one of the most widely used features in audio processing applications like surveillance-related events [2], environmental sound classification [88], speech recognition [89], speech enhancement [90], speaker recognition [91], music genre classification [92], and more.

Feature	Popular Audio-Based Applications	Paper	Domain
STFT Magnitude Spectrum	Respiratory abnormality detection	[82]	Time-frequency domain
STFT Spectrogram	Environmental sound classification	[83]	Time-frequency domain
Mel-Spectrogram	capturing time-frequency characteristics, identify and categorize audio data, speech processing	[84][85][86]	Cepstral Domain
Mel Frequency Cepstral Coefficients (MFCCs)	Surveillance-related events, environmental sound classification, speech recognition, speech enhancement, speaker recognition, music genre classification	[2][88][89] [90][91][92]	Cepstral Domain
Peak frequency	Music and speech classification, gender classification	[75][78]	Frequency Domain
Chroma-based features	Music genre classification	[79]	Frequency Domain
Spectral Centroid	Music classification, music mood classification	[80][81]	Frequency Domain
Zero-crossing rate (ZCR)	Music/speech discrimination, music genre classification, voice activity detection and vowel detection and analysis	[52][53][54] [55]	Time Domain
Amplitude descriptor (AD)	Environmental sound classification	[56]	Time Domain
Log attack time (LAT)	Music genre classification	[58][59]	Time Domain
Shimmer	Stress and emotion classification, speaker detection and verification, music sound classification	[61][62][63]	Time Domain
Short time energy (STE)	Environmental sound detection, audio-based surveillance systems, music onset detection, vowel detection and analysis	[65][66][67] [55]	Time Domain
Volume	Acoustic scene classification, speech and music classification, speech segmentation	[68][69]	Time Domain
Temporal centroid (TC)	Environmental sound classification, acoustic scene classification	[70][71]	Time Domain
Auto-correlation based features	Acoustic scene classification, music tempo and beats estimation	[72][73]	Time Domain
Rhythm-based features	Music genre classification, music instrument classification, speech/music discrimination, analysis of pathological speech	[74][69][93]	Time Domain

Table 6.1.3

6.1.4 Acoustic Localization Methods

Time Difference of Arrival (TDOA)

The TDOA method works by measuring the time differences in sound arrival at multiple microphones to locate the sound source. The key idea is that the sound reaches each microphone at different times depending on its distance from the source.

How It Works:

Microphones are placed at fixed, known locations, usually arranged in a grid or circular pattern. When the drone produces noise, each microphone records the sound at a slightly different time. The system measures these time differences and uses them to calculate the drone's position by applying advanced algorithms.[94]

Phase Array

The Phase Array method analyses phase shifts in the sound waves received by multiple microphones to determine the direction of the sound source. The microphones capture the sound at different phases depending on their position relative to the source.

How It Works:

Microphones are placed in a structured pattern, such as a linear array or grid. As the drone emits sound, each microphone picks up the sound waves with a unique phase. By comparing the phase differences between the microphones, the system estimates the direction of the drone and calculates its location using phase analysis algorithms.[94]

Amplitude

The amplitude-based method for localizing sound sources works by analyzing the differences in sound intensity (amplitude) received by multiple microphones. As sound travels from its source, its intensity decreases, which can be used to estimate the distance of the sound source from each microphone. By comparing the sound levels at each microphone, the system can estimate the location of the sound source.

How It Works:

Microphones are placed at known locations. When the drone makes noise, each microphone records the amplitude of the sound. The closer the microphone is to the drone, the stronger the signal it picks up. By comparing these amplitude differences and using the inverse square law, the system calculates the relative distances from the microphones to the drone and estimates its position. [95]

6.2 Proprietary and business information

6.2.1 Market Survey

1. **Meduza X**- is an advanced drone detection system developed by The Third Eye, employs cutting-edge image processing to accurately identify airborne objects. The system scans the environment in 360° with high-speed using a unique mirror technique, allowing for rapid and complete coverage. Weighing just 12 kg, it is portable and can be carried in a compact bag, facilitating quick deployment in the field. Meduza employs advanced sensors and edge computing, requiring minimal pixels for detection while delivering precise results with a low false alarm rate (FAR).

Figure 6.2.1.1

Figure 6.2.1.2

2. **Drone Dome-** is a system developed by Rafael. It detects and identifies threatening drones using radar and electro-optical sensors. Once a threat is detected, the data is integrated, and an alert is sent to the operator. The system can then initiate jamming either automatically or manually, based on predefined rules. When the threat reaches the neutralization zone, the hostile drone is neutralized by jamming its GPS and radio signals or by being intercepted with a laser system.

Figure 6.2.1.3

3. **Discovair G2**- is an acoustic-based drone detection system that addresses a critical need for detecting drones in proximity, particularly in environments where line-of-sight detection is challenging. It is effective in urban areas, around obstacles, and in conditions like darkness or fog, where traditional radar and optical systems may struggle. The system utilizes patented microphone arrays consisting of 128 interconnected microphone elements. These microphones work together to establish the azimuth and elevation of a target in real-time through advanced digital signal processing. The modular and scalable design of Discovair G2 allows it to be deployed in various configurations, from small mobile units to large perimeter or border defenses.

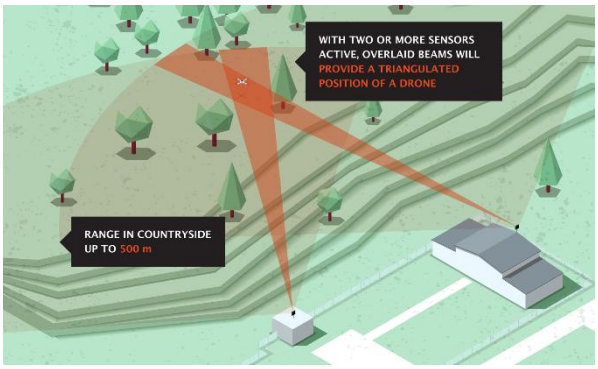


Figure 6.2.1.4

Figure 6.2.1.5

6.2.2 Comparison Table of Existing Solutions

Attribute	Drone Dome	Discovair CUAS	Meduza X
Range	5	4	5
Mobility	6	9	7
Size	5	9	8
Price	3	7	6
Power Consumption	2	9	7
No Line of Sight	0	9	0
Weather Resistance	3	6	3
Final Score	3.4	7.7	5.1

Table 6.2.2

Explanation:

Drone Dome has a low score for range in detecting small autonomous drones due to its reliance on RF and radar technologies, which aren't effective for this purpose. Discovair G2 excels in mobility, low power consumption, and effectiveness in environments without line of sight, making it the best option for detecting small autonomous drones. Meduza X has moderate scores but is limited by poor visibility and line-of-sight requirements, which affect its overall effectiveness.

It is clear that Discovair G2 is the most suitable for this purpose, and in our project, we chose to work in this direction with the goal of creating a similar 'blue white' product to reduce costs.

6.3 Patent Review for Acoustic Drone Detection

In this section, we review three patents related to acoustic drone detection. This analysis will serve to inform the design of our own system while ensuring compliance with existing intellectual property rights.

1. US11594141B1 - Distributed Airborne Acoustic Anti-Drone System

This U.S. patent focuses on a comprehensive system for detecting drones through their acoustic signatures. The system uses **microphone arrays** to capture drone sounds and processes these sounds using **spectrograms** (like Mel-frequency cepstral coefficients, or MFCCs) and **adaptive filters** to isolate relevant acoustic signals.

The distinguishing feature of this patent is its use of **Concurrent Neural Networks (CoNNs)**, which process the acoustic data to classify drone types and predict their movement. The system employs machine learning models, including **Random Forests** and **Multi-Layer Perceptrons (MLPs)**, to ensure accurate detection and differentiation of drone types. This patent particularly focuses on **real-time detection** and **trajectory prediction** based on audio data (MDPI)(SpringerLink).

2. EP3371619B1 - Drone Detection Using Microphone Arrays

This European patent covers methods for spatially detecting drones using **microphone arrays** and **acoustic cameras**. The primary focus here is on the **spatial localization** of drones, achieved through techniques like **beamforming**, which directs the microphone arrays toward suspected drone locations to minimize interference from background noise.

The system analyzes the acoustic signals from drones using **frequency-domain analysis** and passes the processed signals to a machine learning algorithm for classification. **Neural networks** are then used to identify the drone's acoustic signature and predict its position. The patent also describes methods for enhancing detection accuracy through **harmonic-percussive source separation** (MDPI)(SpringerLink).

3. EP1946606B1 - Acoustic and Visual Drone Detection and Trajectory Prediction

This European patent primarily focuses on integrating **acoustic** and **visual data** to detect drones. The patent's core innovation is its method of **trajectory prediction** based on the combination of different detection mechanisms. By fusing acoustic signals with other sensor data (such as visual or radar), the system increases the accuracy of drone detection and tracking.

The acoustic detection system relies on **spectral analysis** to identify drone signatures. The patent also describes a method of **predicting a drone's flight path** by analyzing its detected audio signals over time and correlating these with other sensor inputs(MDPI).

Measures to Avoid Patent Infringement

To ensure our acoustic drone detection system does not infringe on these patents, we must take the following precautions:

1. **Feature Extraction**:
    - **US11594141B1** utilizes **MFCCs** and **log-Mel spectrograms** for feature extraction. To avoid infringement, we can opt for alternative methods such as **wavelet transforms** or **short-time Fourier transforms (STFT)**, or explore different signal processing techniques for acoustic feature extraction.
2. **Signal Classification**:
    - While machine learning models such as MLPs or Random Forests are not inherently protected, the specific way they are applied in conjunction with **acoustic preprocessing** is protected by these patents. We can avoid infringement by designing custom machine learning pipelines and utilizing alternative models, such as **Support Vector Machines (SVMs)** or **Convolutional Neural Networks (CNNs)**, with unique training data.
3. **Spatial Localization**:
    - The beamforming techniques used in **EP3371619B1** are protected, so alternative approaches should be explored for spatial localization, such as using **cross-correlation techniques** or different array processing methods that do not replicate beamforming.
4. **Multimodal Data Fusion**:
    - **EP1946606B1** integrates acoustic and visual data for enhanced detection and tracking. While data fusion itself is not a new concept, care must be taken not to directly replicate the specific fusion techniques outlined in the patent. Alternative sensor fusion methods or focusing on solely acoustic data could provide differentiation.

By implementing alternative algorithms for feature extraction, classification, and detection, we can develop a unique product while avoiding the infringement of existing patents.

7. Methodology

7.1. Components Alternatives

7.1.1 Processor Selection

Here is a brief explanation of each processor under consideration:

Raspberry Pi 4 Model B: An affordable, versatile single-board computer, suitable for lightweight AI tasks using external libraries.

NVIDIA Jetson Xavier NX: Designed for real-time AI processing with powerful GPUs and Tensor Cores, making it ideal for complex neural network applications.

Intel NUC: A compact mini-PC with a range of Intel processors, offering flexibility and decent performance for general computing and AI tasks.

Google Coral Dev Board: Equipped with an Edge TPU for AI at the edge, ideal for machine learning models optimized for low-power, real-time applications.

BeagleBone AI-64: A high-performance board with an AI accelerator, built for efficient processing of AI tasks and suitable for industrial AI applications.

UP Squared Pro AI Edge: A versatile board with support for AI via VPU and FPGA, combining affordability with powerful AI capabilities.

Detailed Comparison:

Category	NVIDIA Jetson Xavier NX	Raspberry Pi 4 Model B	Intel NUC	Google Coral Dev Board	BeagleBone AI-64	UP Squared Pro AI Edge
CPU Speed	6-core NVIDIA Carmel ARMv8.2 1.4GHz **(9)**	Quad-core Cortex-A72 1.5GHz **(8)**	Intel Core i3/i5/i7 (up to 4.5GHz) **(7)**	Quad-core Cortex-A53 1.5GHz **(6)**	Octa-core Cortex-A72 2.2GHz **(6)**	Intel Celeron/Pentium/Atom 2.6GHz **(6)**
GPU	384-core NVIDIA Volta + 48 Tensor Cores **(9)**	VideoCore VI **(7)**	Integrated Intel Graphics **(5)**	Google Edge TPU **(8)**	Integrated PowerVR GPU **(6)**	Intel UHD Graphics **(6)**
Price (USD)	$399 **(6)**	$35 - $75 **(10)**	$300 - $1000 **(5)**	$130 **(7)**	$140 **(7)**	$299 **(6)**
AI Focus	Yes (Optimized for AI) **(10)**	No (AI via libraries) **(6)**	No (Can run AI apps) **(5)**	Yes (Edge TPU for AI) **(8)**	Yes (AI accelerator) **(7)**	Yes (Supports AI with VPU/FPGA) **(7)**
OS Support	Ubuntu, NVIDIA JetPack **(9)**	Raspberry Pi OS, Linux, Windows IoT **(8)**	Windows, Linux **(7)**	Mendel Linux, Debian **(7)**	Linux (Debian-based) **(6)**	Ubuntu, Windows, Yocto Linux **(6)**
Final Score	**43**	**39**	**30**	**36**	**35**	**30**

Table 7.1.1

Best Choice for Real-Time AI Processing

The NVIDIA Jetson is the best choice for our project due to its powerful GPU optimized for AI tasks, support for neural networks, and real-time image and sound processing. It is highly recommended for intensive AI applications.

Budget Consideration

We will initiate development using Raspberry Pi 4, as it offers an affordable and versatile foundation for our project. However, if we determine that its processing power is insufficient for our computational requirements, we will transition to Nvidia Jetson, which provides enhanced performance capabilities suitable for more demanding tasks.

7.1.2 Microphone Selection

There are several types of microphones, each suited for different recording environments and needs:

- **Dynamic Microphones**: Rugged and can handle high sound pressure levels, making them ideal for live performance and noisy environments.
- **Condenser Microphones**: More sensitive and provide a broader frequency response, making them suitable for studio recordings and capturing fine details.
- **Directional Microphones**: Focus on picking up sound from a specific direction, crucial for drone detection.

Types of Directional Microphones

- **Parabolic Microphones:** These use a parabolic dish to focus sound waves onto a central microphone. This results in enhanced capture of distant sounds while reducing noise from the sides. Parabolic mics are ideal for long-range applications, such as wildlife recording or surveillance.
- **Shotgun Microphones**: Feature a narrow, elongated design that focuses on sound directly in front of them while rejecting sound from the sides and rear. They are portable and well-suited for medium-range applications.
- **Microphone Array:** Systems of multiple microphones arranged to capture sound dynamically across multiple directions. They offer high directivity and noise rejection, making them ideal for long-distance sound capture, though at a higher cost and complexity.

Comparison Table: Parabolic vs. Shotgun Microphones

Feature	**Parabolic Microphone**	**Shotgun Microphone**	**Microphone Array**
Sensitivity	Highest sensitivity **(10)**	Moderate sensitivity **(8)**	High sensitivity **(9)**
Frequency Response	Depending on the size **(8)**	Excellent **(9)**	Depending on the size **(8)**
Size	Bulky and larger **(6)**	Compact and portable **(9)**	Medium size **(7)**
Directivity	Excellent, Static **(9)**	Good, Static **(8)**	Excellent, Dynamic **(10)**
Range	Extremely High **(10)**	Medium **(7)**	High, limited by noise-floor **(9)**
Suitability	Best for long-distance sound **(10)**	**Best for medium-range sound (8)**	**Good for long-distance sound (9)**
Noise Rejection	Excellent **(10)**	Good **(7)**	Excellent **(10)**
Cost	Expensive **(6)**	Affordable **(9)**	Very Expensive **(5)**
Final Score	**69**	**65**	**67**

Table 7.1.2.1

Conclusion:

The parabolic microphone is the better choice due to its superior range and ability to isolate drone sounds. However, parabolic microphones are typically more expensive and larger, making them less portable and harder to handle in some situations.

Due to the high cost and difficulty in finding a parabolic microphone that meets our specific requirements, we have decided to develop a custom array of parabolic microphones specifically for this project. The development process will be detailed later in this document. For developing a parabolic microphone, we need to choose omnidirectional microphone to mount on the parabolic deflector. Below is a comparison table for several types of omnidirectional microphones:

Category	**Saramonic SR-LMX1+**	**Rode Lavalier GO**	**Shure MVL**	**Boya BY-M1**	**Audio-Technica PRO 70**
Price (USD)	$49 **(7)**	$79 **(6)**	$69 **(6)**	$20 **(8)**	$129 **(5)**
Sensitivity (±dB)	-30 ±3 **(8)**	-35 ±1 **(7)**	-44 ±2 **(6)**	-30 ±4 **(7)**	-43 ±2 **(6)**
SNR (dB)	74 **(9)**	67 **(6)**	65 **(6)**	74 **(8)**	72 **(6)**
Frequency Response (Hz)	30 – 18,000 **(8)**	20 – 20,000 **(10)**	45 – 20,000 **(7)**	65 – 18,000 **(7)**	50 – 15,000 **(6)**
Final Score	**32**	**30**	**25**	**31**	**25**

Table 7.1.2.2

Conclusion:

Additionally, we need to find a microphone that does not have built-in processing or one where we can disable it, as some microphones with internal processing might filter out or ignore the specific drone sounds we are trying to capture.

Among the omnidirectional microphones listed, the **Saramonic SR-LMX1+** was chosen for its excellent balance of performance and cost. It captures subtle sounds effectively and provides reliable quality, making it a practical and affordable option for this project.

As a backup plan, we will use shotgun microphones, which, while not as effective as parabolic ones for this purpose, still meet the project's performance standards and can serve the project's needs. Below is a comparison table for different types of shotgun microphones:

Category	**Takstar SGC-598**	**BOYA BY-MM1**	**Movo VXR10**	**Rode VideoMicro**	**Comica CVM-V30 PRO**
Price (USD)	$35 **(8)**	$40 **(7)**	$39 **(7)**	$35 **(9)**	$58 **(6)**
Sensitivity (dB)	-32 **(8)**	-42 **(8)**	-42 ±3 **(7)**	-33 ±2 **(9)**	-40 ±2 **(6)**
SNR (dB)	74 **(8)**	78 **(9)**	78 **(9)**	74 **(8)**	75 **(9)**
Frequency Response (Hz)	50 – 16,000 **(7)**	35 – 18,000 **(9)**	35 – 18,000 **(9)**	100 – 20,000 **(9)**	40 – 20,000 **(9)**
Final Score	**31**	**31**	**31**	**33**	**26**

Table 7.1.2.3

Conclusion:

Among the shotgun microphones listed, the **Rode VideoMicro** is the best choice for the project due to its good balance of price, sensitivity, and frequency response. It will provide clearer recordings over medium distances.

7.2 Project Planning

To achieve the goal of detecting drones at specific distances, it is essential to understand how sound attenuates in space, calculate its intensity at the target distance, and design a parabola that provides sufficient amplification for the microphone to capture the sound. Additionally, considering that we are dealing with a parabolic microphone with a non-uniform frequency response, and that at large distances lower frequency ranges might not be captured by dishes that are not large enough, it is necessary to first analyze the drone's behavior across different frequency ranges to ensure detection feasibility.

Subsequently, the model will be adjusted to recognize similar frequency behavior. This process includes understanding the frequency ranges and their attenuation as a function of distance, designing a parabolic dish to provide the required amplification, analyzing the expected frequency distortion, developing a software filter to compensate for this distortion, and passing the data through the filter to train the model on the expected frequency behavior.

7.2.1 Drone Frequency Behavior

In a drone, each motor has a fundamental frequency related to its rotational speed, expressed as:

$$F_{motor} = \frac{RPM}{60}$$

Because each propeller has multiple blades, a blade-passing frequency (BPF) also appears as:

$$F_{BPF} = N_{blade} \cdot F_{motor}$$

Neither the motor nor the blade-passing events are purely sinusoidal, so each one generates harmonics at integer multiples of its base frequency. In a multi-rotor system, each motor/propeller assembly contributes its own blade-passing frequency, meaning the total number of blade-passing events becomes:

$$F_{BPE} = N_{motor} \cdot N_{blade} \cdot F_{motor}$$

This can introduce additional discrete frequencies in the noise spectrum, especially when motors operate at nearly the same speed. Consequently, the harmonic profile is richer than in a single-rotor case, and certain frequencies may be amplified further by structural resonances or constructive interference among the rotors.

When these harmonics combine, they can either reinforce each other (constructive interference) or cancel each other out partially (destructive interference). A common way to estimate their amplitude decay is the power-law relationship:

$$A_k = \frac{A_1}{k^n}$$

where:

- A_1 is the fundamental amplitude
- k is the harmonic index
- n typically ranges from 1 to 2.

Figure 7.2.1.1 is an FFT representation of a DJI mini 3 PRO with 4 motors and 2 blades for each.

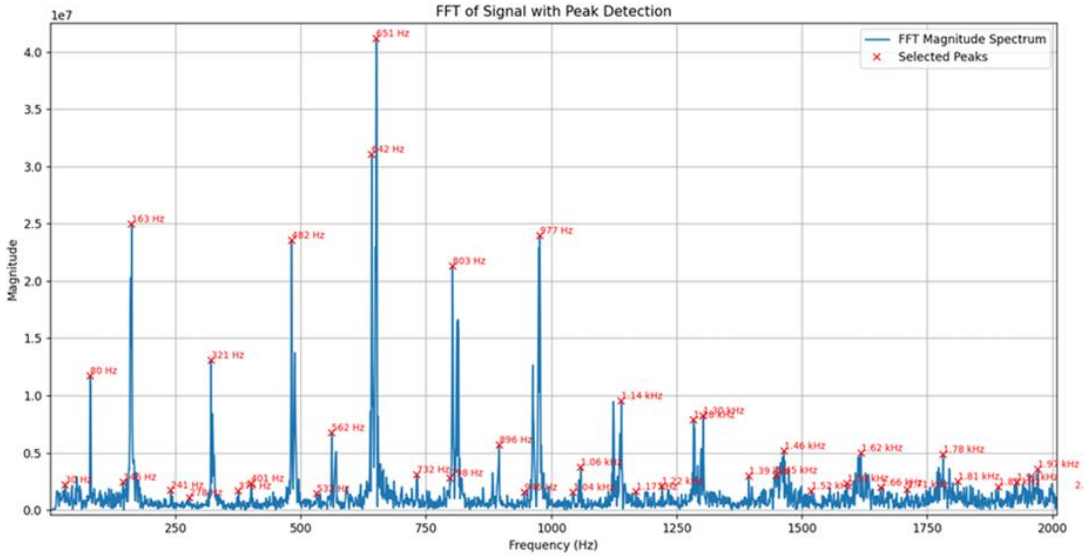


Figure 7.2.1.1

What we see in the FFT graph:

- **80Hz:** 1st peak, this is the base frequency as a result of the motor spin. We expect harmonics at 80Hz, 160Hz, 240Hz, 320Hz, and so on.
- **160Hz:** 2nd peak, this results from the base frequency multiplied by the number of blades (80 * 2 = 160). We expect harmonics at 160Hz, 320Hz, 480Hz, and so on.
- **640Hz:** This peak results from the blade-passing events across all motors, $(80 \cdot 4 \cdot 2 = 640)$. We expect harmonics at 640Hz, 1280Hz, 1920Hz, and so on.

  In certain cases, the harmonics of different base frequencies can overlap, creating points where the amplitudes reinforce each other, resulting in higher intensity at those frequencies. For example, at 1280 Hz, the harmonics of the motor frequency (80 Hz), blade-passing frequency (160 Hz), and combined blade-passing events frequency (640 Hz) align. This happens because 1280 Hz is a common multiple of all three base frequencies, leading to constructive interference and a significantly stronger harmonic at this frequency.

Figure 7.2.1.2 is an FFT representation of the drone's higher frequencies. It can be observed that the harmonics are still identifiable at the expected locations where they are predicted to appear.

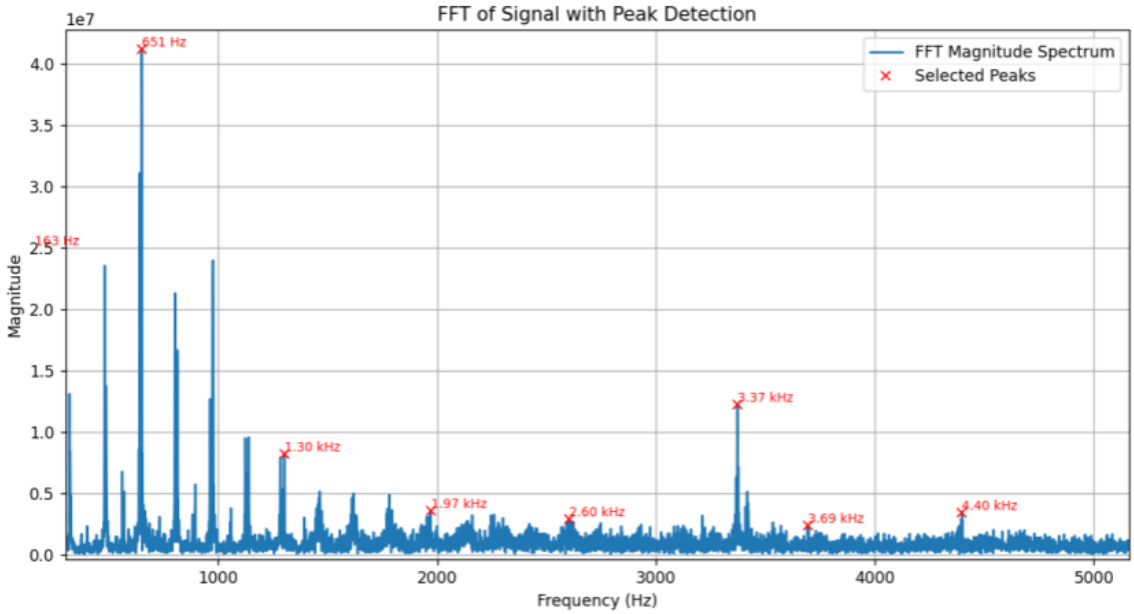


Figure 7.2.1.2

Conclusion

Based on the above explanation, it can be theoretically assumed that a drone can still be identified by relying solely on its higher harmonies. These higher frequencies, created by the overlap and reinforcement of various harmonics, retain a distinct spectral signature even if the lower frequencies are filtered out.

7.2.2 Sound Attenuation in the air

Introduction

When a sound source emits acoustic energy in a free field (open area without significant reflections), the sound pressure level (SPL) decreases with distance due to:

1. **Geometrical spreading** (inverse-square law).
2. **Atmospheric absorption**, which is frequency-dependent and generally more pronounced at higher frequencies.

Geometrical spreading

For a point source radiating uniformly in all directions, the drop in level in dB moving from a reference distance d_1 to a new distance d_2 is:

$$L_2 = L_1 - 20log(\frac{d_2}{d_1})$$

Where:

- d_1: Reference Distance, commonly set to 1m.
- d_2: Distance from the sound source.
- L_1: The sound intensity at distance d_1.
- L_2: The sound intensity at distance d_2.

Distance [m]	25	100	500
Magnitude drop	-28db	-40db	-54

Table 7.2.2.1

Atmospheric Absorption

Atmospheric absorption arises mainly from molecular relaxation processes in the air, highly dependent on frequency, temperature, and relative humidity. This absorption can be approximated by:

$$L_{(r)} = L_0 - \alpha_{(f)} \cdot (d - d_0)$$

where $\alpha_{(f)}$ is the absorption coefficient for frequency f (in dB per meter or dB per 100 meters), and d is the distance traveled.

The attenuation of sound in air is characterized by the attenuation coefficient $\alpha_{(f)}$, which depends on the frequency f, atmospheric pressure, relative humidity, and temperature. The general formula for $\alpha_{(f)}$ is given by:

$$\alpha(f) = 8.686 \cdot f^2 \cdot (\frac{\eta_{O_2}}{f^2 + f_{O_2}^2} + \frac{\eta_{N_2}}{f^2 + f_{N_2}^2})$$

Where:

- f: Frequency (Hz).
- f_{O_2}, f_{N_2}: Relaxation frequencies of oxygen and nitrogen, respectively.
- $\eta_{O_2}, \eta_{\mathrm{N}_2}$: Absorption coefficients for oxygen and nitrogen, respectively.

Attenuation Coefficient Calculation

Environmental Parameters:

For the calculations, we assume typical environmental conditions:

- **Barometric Pressure**: $p = 101{,}325Pa$
- **Temperature**: $T = 293.15K\ (20°)$.
- **Relative Humidity**: $h = 0.5\ (50\%)$.

Relaxation Frequencies:

The relaxation frequencies f_{O_2} and f_{N_2} are calculated as follows:

Oxygen Relaxation Frequency (f_{O_2}):

$$f_{O_2} = \frac{p}{p_0} \cdot \left(24 + 4.04 \cdot 10^4 \cdot h \cdot \frac{0.02 + h}{0.391 + h}\right) = 24 + 4.04 \cdot 10^4 \cdot 0.5 \cdot \frac{0.52}{0.891} \approx 11{,}788Hz$$

Nitrogen Relaxation Frequency (f_{N_2}):

$$f_{N_2} = \frac{p}{p_0} \cdot \left(9 + 280 \cdot h \cdot e^{-4.17 \cdot \left(1 - \frac{T}{T_0}\right)}\right) = 9 + 280 \cdot 0.5 \cdot 1 = 149Hz$$

Attenuation Coefficient Formula:

After substituting these values into the general formula for $\alpha_{(f)}$, we obtain:

$$\alpha(f) = 8.686 \cdot f^2 \cdot \left(\frac{0.012}{f^2 + (11{,}788)^2} + \frac{0.00013}{f^2 + (149)^2}\right)$$

And when substituting the coefficient:

$$L_{(r)} = L_0 - \alpha_{(f)} \cdot (d - d_0) = L_0 - 8.686 \cdot f^2 \cdot \left(\frac{0.012}{f^2 + (11{,}788)^2} + \frac{0.00013}{f^2 + (149)^2}\right) \cdot (d - d_0)$$

To simplify spectral analysis, we often group frequencies into octave bands. Approximate center frequencies (Hz) for the range 50–8,000 Hz might be:
63 Hz, 125 Hz, 250 Hz, 500 Hz, 1 kHz, 2 kHz, 4 kHz, 8 kHz.

Example Atmospheric Absorption at different frequencies

Octave Center	Approx. α (dB per 100 m)	Magnitude drop at 25m	Magnitude drop at 100m	Magnitude drop at 500m
63 Hz	0.01	-0.004db	-0.017db	-0.087db
125 Hz	0.01	-0.011db	-0.047db	-0.238db
250 Hz	0.02	-0.021db	-0.087db	-0.439db
500 Hz	0.05	-0.029db	-0.121db	-0.611db
1 kHz	0.10	-0.044db	-0.183db	-0.923db
2 kHz	0.20	-0.097db	-0.4db	-2.016db
4 kHz	0.60	-0.285db	-1.177db	-5.933db
8 kHz	2.00	-0.816db	-3.366db	-16.965db

Table 7.2.2.2

Putting It All Together

By combining the formula for spherical spreading and atmospheric absorption, we obtain:

$$L_{(f,d)} = L_1 - 20\log\left(\frac{d}{d_1}\right) - d \cdot \alpha_{(f)} =>$$

$$=> \boldsymbol{L_{(f,d)} = L_1 - 20\log\left(\frac{d}{d_1}\right) - d \cdot 8.686 \cdot f^2 \cdot \left(\frac{0.012}{f^2 + (11,788)^2} + \frac{0.00013}{f^2 + (149)^2}\right)}$$

This equation provides an accurate description of sound intensity at varying distances as a function of frequency, incorporating the effects of spherical spreading and high-frequency absorption in the air.

Below is a final calculation of magnitude drop for the combined attenuations:

Octave Center	Magnitude drop at 25m	Magnitude drop at 100m	Magnitude drop at 500m
63 Hz	-27.963db	-40.017db	-54.066db
125 Hz	-27.97db	-40.047db	-54.217db
250 Hz	-27.98db	-40.087db	-54.418db
500 Hz	-27.988db	-40.121db	-54.59db
1 kHz	-28.003db	-40.183db	-54.902db
2 kHz	-28.056db	-40.4db	-55.995db
4 kHz	-28.244db	-41.177db	-59.912db
8 kHz	-28.775db	-43.366db	-70.944db

Table 7.2.2.3

Practical calculation for our needs

DJI Air 2s	FUNSKY 913	DJI Mini 2	DJI Mavic 3	Altair Aerial AA108	DJI Air 3	DJI Mini 3 Pro
64 db	68 db	64 db	69 db	65 db	65 db	70 db

Table 7.2.2.4

From the table above, we can observe the various types of quieter drones, which reach a minimum sound level of 64 dB. We will set 60 dB as the lower limit for our analysis.

for 60db as our lower limit:

Distance (m)	63 Hz	125 Hz	250 Hz	500 Hz	1 kHz	2 kHz	4 kHz	8 kHz
25 m	32	32	32	32	32	31.9	31.7	31.2
100 m	20	20	19.9	19.9	19.8	19.6	18.8	16.6
500 m	5.9	5.8	5.6	5.4	5.1	4	0.1	-10.9

Table 7.2.2.5

From the table above, we understand that in the worst case, we observe a signal level of -10.9 dB at a distance of 500 meters and a frequency of 8 kHz. This indicates that signals at this frequency and distance are significantly attenuated, making detection challenging. To address this, we need to optimize our system's sensitivity and take it into consideration.

7.2.3 Parabolic Reflector design

To detect a drone at a distance of 25 meters, the sound intensity must be amplified near the microphone. The microphone used in our project has a sensitivity of -30 dB. We aim for a gain that exceeds the microphone sensitivity by at least 10 dB to ensure a reliable signal. Therefore, we designed the dish to provide a minimum gain of 8 dB to ensure a clearer sound.

Gain

A parabolic dish is used to enhance the sound signal further by focusing acoustic energy toward the microphone. The gain provided by the dish improves the detection range and clarity of the drone's acoustic signal.

The gain of the dish is calculated using the formula:

$$G = 10 \cdot log_{10}((\frac{\pi D f}{c})^2 \eta)$$

Cutoff Frequency

The cutoff frequency is the point at which the dish starts to provide positive gain and being efficient. It can be calculated using the following formula:

$$f_{cut-off} = \frac{c}{\pi \cdot D}\sqrt{\frac{2}{\eta}}$$

Where:

- D – Dish diameter (meters)
- c– Speed of sound in air (343 m/s)
- η – Dish efficiency (typically 0.6)

Dish Size Selection

We chose a parabolic dish with a diameter of 33 cm, which provides a cutoff frequency of approximately 604.3 Hz. This cutoff aligns well with the frequency ranges that are relevant to our application, ensuring the dish effectively amplifies the relevant sound frequencies. The chosen size offers a balanced solution, providing sufficient gain above the cutoff while maintaining practicality in terms of dish size and deployment.

The following table shows the calculated gain of the chosen dish at specific frequencies:

Frequency (Hz)	200	500	800	1131	1600	2262	3200	4525	6400	9050
Gain (dB)	0	1.37	5.45	8.46	11.47	14.48	17.49	20.5	23.51	26.52

Table 7.2.3.1

It can be observed that the gain becomes positive near the cutoff frequency. Beyond this point, the dish provides a gain of at least 8dB starting at approximately 1kHz, with further increases at higher frequencies.

Beamwidth

Beamwidth refers to the angular width of the sound detection area provided by the parabolic dish. At lower frequencies, the beamwidth is wide, allowing the dish to capture sound from a broader area. However, as frequency increases, the beamwidth narrows, focusing the detection on a smaller area. This narrowing of the beamwidth is directly influenced by the size of the dish, larger dishes result in narrower beamwidths.

The Beamwidth of the dish is calculated using the formula:

$$\theta = 70 \cdot \frac{\lambda}{D}$$

Where:

- θ - Half-power beamwidth (in degrees)
- $\lambda = \frac{c}{f}$ – Wavelength (meters)

We calculated the beamwidth for the chosen dish across various frequencies:

Frequency (Hz)	200	500	800	1131	1600	2262	3200	4525	6400	9050
Beamwidth (degrees)	360	145.50	90.95	64.31	45.47	32.15	22.74	16.08	11.37	8.04

Table 7.2.3.2

From the table, it is evident that as frequency increases, the beamwidth narrows, providing a focused detection area. This consideration was a critical part of our design decision. While larger dishes offer increased gain, they also narrow the beamwidth excessively, which could limit the detection area.

Focal Point and f/D Ratio

The focal point is the specific location where sound waves reflected by the parabolic dish converge. This is the spot where the microphone element should be mounted for optimal sound capture.

The focal point of the dish is calculated using the formula:

$$F = (\frac{D}{16z})^2$$

Where:

- F – is the focal length,
- Z – is the depth of the dish.

The f/D ratio represents the relationship between the distance to the focal point and the dish diameter. It has a significant impact on the microphone's polar pattern. A lower f/D ratio favors a wider polar pattern, allowing the microphone to pick up sound from the entire dish surface. This ensures efficient sound concentration while minimizing off-axis noise.

In contrast, a higher f/D ratio results in a narrower polar pattern and places the focus outside the dish’s mouth. In such cases, the microphone would pick up ambient noise from the sides of the dish, reducing the system's efficiency and increasing interference. Additionally, higher f/D ratios make the system bulkier and require larger structures to position the microphone correctly.

We selected an f/D ratio of 0.25, which represents a practical balance. This ratio ensures that the microphone remains within the dish’s boundaries. The chosen ratio provides a compact design while maintaining efficient sound capture and minimizing interference.

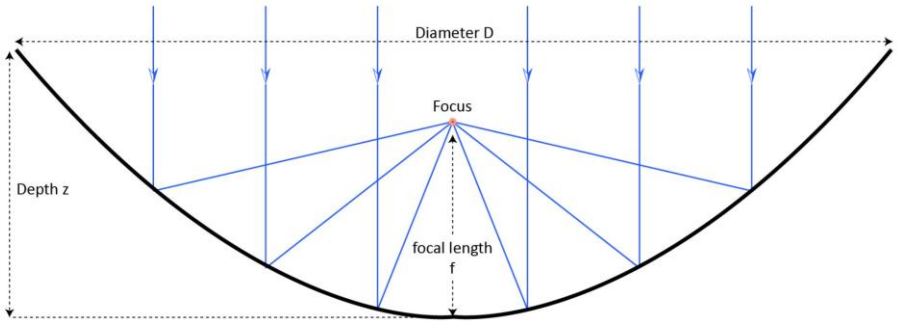


Figure 7.2.3.1

7.2.4 Experiment Procedure Design for the Parabolic Reflector

Given that commonly available simulations are often designed for vacuum environments and do not account for the behavior of sound waves in air, it was decided to conduct a practical experiment. As we do not have access to a sufficiently large acoustic chamber, the experiment will take place outdoors to minimize sound reflections. The chosen location is a large park far from roads and significant noise sources, allowing for a cleaner acoustic environment. However, we will still need to employ advanced noise filtering techniques to ensure accurate results by isolating unwanted noise from the recordings.

Objective

The primary goal of this experiment is to:
1. Analyze the gain of the parabolic microphone across different frequencies.
2. Examine its directionality and determine its effective beamwidth.
3. Derive the frequency response of the parabolic dish to adapt the model's parameters for optimal performance.

Experimental Design

A sound source will be placed at a reasonable distance from the microphone, playing audio files containing one or two frequencies per octave. Due to wave interference and noise filtering considerations, each frequency will be played individually.

The sound will be recorded with and without the parabolic dish. The output amplitudes will then be compared to calculate the gain provided by the dish without considering the microphone's inherent frequency response. Recordings at different angles will also be performed to analyze the frequency response and estimate the dish's effective beamwidth.

Result Analysis

We anticipate the presence of noise in the recordings. However, unlike the desired frequency, which will be consistently present throughout the recording, noise will appear randomly. To address this, the following method has been proposed:
- Record 3 seconds for each frequency.
- Divide each recording into 30 segments of 100 milliseconds.
- Perform a Fourier transform on each segment individually.
- Calculate the median amplitude at the desired frequency across all 30 transforms, smoothing out random peaks caused by noise.

In figure 7.2.4.1: The prominent red line represents the median. It is evident that outside the boundaries of the peak, the line is nearly smooth and less sensitive to noise, as expected. This characteristic is anticipated to assist in more accurate analysis of the results.

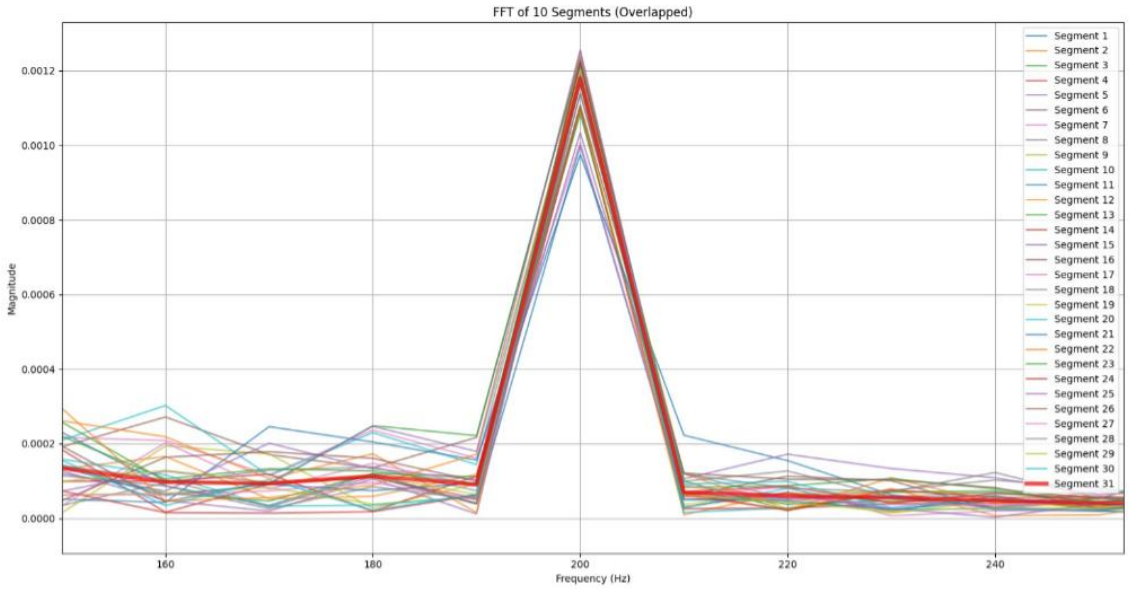


Figure 7.2.4.1

Additional Experimental Tools

A custom experimental system has been developed:

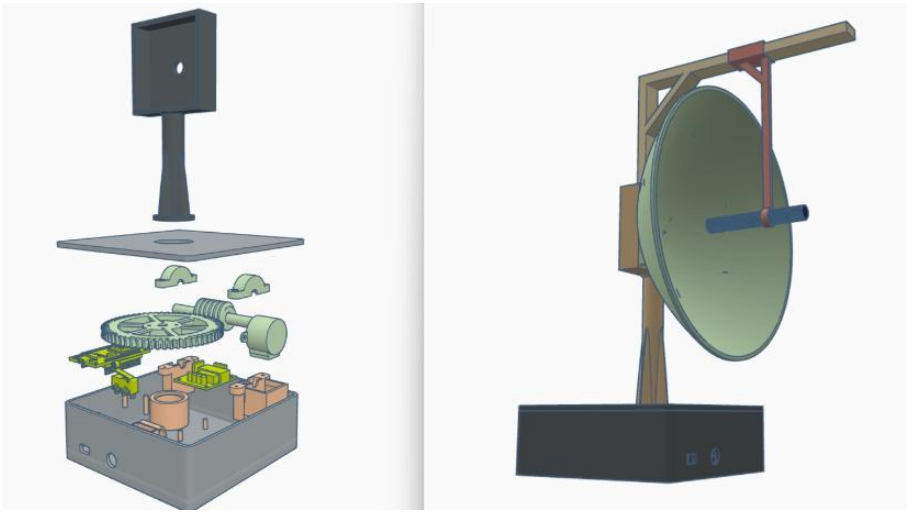


Figure 7.2.4.2

- Automates the movement of the microphone across different angles.
- Synchronizes the sound playback and recordings.
- Organizes and labels the recordings systematically.
- Supports wireless operation and is battery-powered, making it ideal for outdoor use.

Expected Outcome

The experiment aims to produce a graph representing the real-world frequency response and polar pattern of the parabolic dish. This data will serve as a foundation for adapting and optimizing the system to achieve its intended functionality.

7.2.5 Adapting the Data to the Parabolic Reflector

Two critical considerations must be addressed:
first, determining whether the model should be trained on frequencies below the dish's cut-off frequency, and second, how to handle the distortion introduced by the dish at varying angles so the model generalizes drone behavior rather than strictly following specific frequency responses. To address these challenges, two main approaches are considered. The first involves using data manipulation techniques, and the second involves creating a large dataset recorded through the parabolic dish. However, to avoid tying the model too closely to our specific parabolic dish, we will initially emphasize data manipulation methods, thereby ensuring a more generalized system.

Normalization and Standardization

We will implement multiple strategies to control amplitude variations and ensure that no single frequency band disproportionately influences the model. Some commonly used approaches include:

- **Min-Max Normalization:** Scales each frequency or feature to a specific range. This is helpful when data has relatively stable minimum and maximum values and is not heavily affected by outliers.
- **Standard Scaling:** Centers the data by subtracting the mean, then scales it by dividing by the standard deviation. This is effective when the data is approximately normally distributed.
- **Per-Channel Energy Normalization (PCEN):** A more advanced approach often used in audio processing, PCEN adaptively normalizes each frequency band to handle large dynamic ranges. Unlike fixed scalers, it incorporates a smoothing mechanism and can help retain important transient information.
- **Cepstral Mean and Variance Normalization (CMVN):** Commonly applied to Mel-Frequency Cepstral Coefficients (MFCCs), this technique normalizes the cepstral coefficients by subtracting the mean and dividing by the standard deviation within a given time window. This helps mitigate channel or recording environment effects.

Data Augmentation

To broaden the range of training examples and better simulate real-world conditions, we will employ multiple augmentation techniques. By exposing the model to several variations of the same data, we help it become more robust to differences in volume, ambient noise, and microphone sensitivity. The most relevant methods for our parabolic drone-detection setup include:

- **Random Variations in Signal Gain**
  Randomly boosting or attenuating the entire signal's amplitude helps the model cope with different recording levels and slight hardware discrepancies.

- **Frequency Masking**
  Temporarily "blurring out" specific frequency bands in the spectrogram pushes the model to focus on more global features rather than overfitting to narrow frequency cues.

- **Time Masking**
  Randomly zeroing out time segments prevents over-reliance on short-duration events, making the model more tolerant of gaps or missing data.

- **Additive Noise**
  Injecting various noise types (e.g., environmental, mic self-noise) teaches the model to handle background interference and non-ideal conditions.

- **Pitch / Speed Shifting**
  Slightly changing the pitch or playback speed simulates realistic variations in drone rotor RPM or recording conditions, improving generalization.

Cut-off Frequency Filtering

Because the dish itself does not effectively capture frequencies below its cut-off, those rows in the spectrogram will be removed. This ensures that the model isn't trained on data that wouldn't be present in real use cases.

Experimental Method

We will also explore an unvalidated yet promising approach to virtually "warp" the data as if it were recorded through the dish. After measuring the dish's frequency response, we will create a digital filter that emulates how the dish reshapes incoming signals. If this technique proves effective, it will allow us to adapt the model quickly to new microphones or different setups simply by using an appropriate filter matching each hardware configuration, thereby reducing the need for large, dish-specific datasets.

Work Plan

The development of a machine learning model is an evolving process driven by experimentation. At this stage, we do not rely on a single predefined method; instead, we plan to explore various approaches and adapt our strategy based on the outcomes of our experiments. This flexible and iterative process will allow us to refine and optimize the model as we progress.

7.2.6 Choosing the Appropriate Features and Sample Duration

First, we want to know what the parameters are of different drones type for us to take in consideration for choosing the appropriate

Drone Model	RPM	F [Hz]	T [ms]
MINI 3 Pro	3000-10700	50-180	5.5-20
MAVIC 2 Pro	6000-12000	100-200	5-10
FPV	Up to 30,000	500	2

Table 7.2.6.1

We'll take the worst case with the longer time period being 20ms, because of the factor of safety we'll take a larger time period- 100ms.

Our goal is to achieve a sampling duration of 100ms for our model. However, as noted in an article[96], the recommended sample duration is 1 second. Therefore, we will begin by using 1-second recordings as our sample input for the model. If the model performs successfully with this duration, we will gradually reduce the sampling duration, aiming to reach the lower threshold we have set at 100ms.

Once the sampling duration is determined, the next step is to identify the appropriate features to apply to our model. As discussed in Chapter 6.1, we explored various potential features that could be integrated into the system's input. However, after conducting further research and gaining a deeper understanding of the drone's signal and frequency characteristics, we can now pinpoint the features that are most suitable for our needs.

Since we are dealing with various drones that exhibit significant differences in frequency and magnitude characteristics, we will exclude features that rely on magnitude or specific frequencies. In real-world scenarios, noise is inevitable, however, we also know that the drone's signal is continuous throughout the entire sampling duration. Therefore, we will focus primarily on time-frequency domain features. Among these, we have selected three key features for further analysis: Short-Time Fourier Transform (STFT), Mel-Spectrogram, and Mel-Frequency Cepstral Coefficients (MFCC).

Short-Time Fourier Transform (STFT)

The Short-Time Fourier Transform (STFT) is a technique used to analyze a signal in both the time and frequency domains simultaneously. It involves dividing the signal into smaller overlapping time segments (also known as windows) and then applying the Fourier Transform to each segment. This results in a time-frequency representation of the signal, where each point represents the frequency content at a specific time.

The choice of the window size in STFT plays a critical role; smaller windows provide better time resolution but poorer frequency resolution, while larger windows offer better frequency resolution at the expense of time resolution. STFT is widely used in audio signal processing to visualize how frequency components of a signal change over time, making it particularly useful for detecting time-varying features in audio data, such as drone noise or other dynamic sound patterns.

Mel Spectrogram and MFCC

First, the audio is divided into short, often overlapping frames. We then perform the Short-Time Fourier Transform (STFT), which applies a Fourier transform (usually via FFT) to each frame and yields a time–frequency representation. Next, the resulting frequency-domain data is passed through Mel-scaled filter banks (often followed by a logarithmic transform) to produce the Mel spectrogram. Finally, applying the Discrete Cosine Transform (DCT) to these log Mel energies yields the Mel-Frequency Cepstral Coefficients (MFCCs).

Feature	Advantages	Disadvantages	Complexity
STFT	Simple to implement, provides time-frequency analysis, widely used	Trade-off between time and frequency resolution (depends on window size), sensitive to noise	low mathematical complexity, large amount of data generated due to the time-frequency representation
Mel-Spectrogram	Aligns with human auditory perception, effective for speech and environmental sound analysis	Trade-off between time and frequency resolution (due to STFT), requires additional processing (e.g., log scaling)	low mathematical complexity once Mel scale is applied, relatively compact representation compared to STFT
MFCC	Compact representation of audio, robust to noise, highly interpretable	Ignores phase information, sensitive to variations in recording conditions	low to moderate mathematical complexity (logarithms, DCT), Minimal data representation (few coefficients)

Table 7.2.6.2

Conclusion:

- **Real-Time Efficiency:** Since STFT can be computationally heavy, we favor MFCC and the Mel spectrogram. Their more compact representation requires fewer model parameters, making them more suitable for real-time processing.

- **Noise Suppression:** The log transform, an integral part of both Mel spectrogram and MFCC extraction, naturally reduces noise and strengthens drone detection, thus minimizing additional preprocessing.

- **Perceptual Relevance:** Mel spectrograms closely reflect human auditory perception, creating a direct link between what we hear as a drone and what the model is designed to identify. Therefore, we will start with the Mel spectrogram.

7.2.7 Detailed Explanation for the Chosen Features

STFT

The first step is to frame the signal, in which the signal is segmented into overlapping frames. The second step is to apply a window function to prepare and smooth the signal to compute Discrete Fourier Transform (DFT). This step helps to minimize the signal discontinuities at the beginning and end of each frame. A typical window is a Hamming window. The third step is to use DFT to convert the waveform to the spectrogram, which is also a conversion from time domain feature to frequency domain signals. For each k, DFT is defined as follows:

$$\hat{\mathrm{x}}(k) = \sum_{n=0}^{N-1} x(n) e^{-2\pi i \frac{kn}{N}}$$

where k is the index of DFT output in the frequency domain. And $k = -\frac{N}{2}, \dots, -1,0,1, \dots, \frac{N}{2} - 1$. $\hat{x}(k)$ is k^{th} DFT output component. n is the time domain index of input samples, and n = 0, 1, ..., N - 1. x(n) is the discrete sequence of the original sound signal input. N is the number of data samples in the discrete time domain and the number of bins in the discrete frequency domain.

The calculation of the Discrete Fourier Transform (DFT) is done using a Fast Fourier transform. The power spectrum $ps(f)$ is defined as follows:

$$ps(f) = \sqrt{Re\left(H_f\right)^2 + Im\left(H_f\right)^2}$$

At this point, we have successfully generated the STFT feature. This feature can be utilized as a standalone representation, as highlighted in the literature review (Chapter 6.1), where its effectiveness for time-frequency analysis in various applications was discussed. Alternatively, the STFT feature can serve as a foundational element for deriving additional features, as outlined in the subsequent sections below:

Mel-Spectrogram

To calculate the Mel-Spectrogram we apply the Mel-filter bank, which is a set of triangular filters is applied to the power spectrum. These filters are spaced according to the Mel scale, which is designed to replicate the frequency perception of the human auditory system.

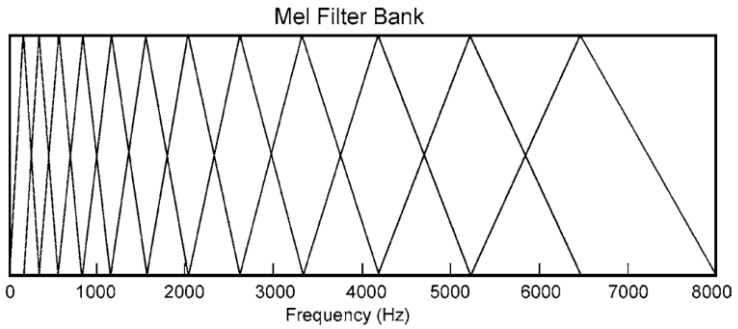


Figure 7.2.7

The Mel scale is calculated as:

$$m_{(f)} = 2595 \cdot \log_{10}(1 + \frac{f}{700})$$

Where f is the frequency in Hz, and $m_{(f)}$ is the corresponding frequency in Mel.

The number of Mels indicates the energy in each filter bank. The purpose of applying Mel scale is that to mimic how human hearing precepts sound.

And now, we have successfully generated the Mel-Spectrogram feature. As previously mentioned, this feature can be used as a standalone representation, providing a perceptually relevant time-frequency analysis. Alternatively, it can serve as a basis for deriving an additional feature:

MFCC

The final step of the process of getting the final feature is to calculate the cepstral coefficients based on the previous step, using Discrete Cosine Transformation (DCT). The calculation of cepstrum is defined as following:

$$c_d = \frac{1}{M}\sum_{m=0}^{M-1} C_m \cos\left(\frac{\pi(2d+1)m}{2M}\right)$$

In the above formula:

- c_d is the d^{th} cepstral coefficient
- M is the total number of filter banks.
- C_m indicates the log energy for filter bank m.

7.2.8 Choosing MEL-spectrogram parameters

Sampling Rate

This sampling rate is typically sufficient to capture the relevant frequency range of drone engines and propellers, as determined earlier (up to the Nyquist frequency of 8 kHz). Since our system is designed to incorporate multiple microphones, using a higher sampling rate for each microphone would result in a significantly larger volume of data to process. This, in turn, could slow down the overall processing time and affect system efficiency. Therefore, selecting an appropriate sampling rate ensures a balance between capturing the necessary frequency information and maintaining computational efficiency.
Currently we've chosen **16KHz** as a starting point.

Feature parameters

When selecting parameters for time-frequency features, it is important to note that the features are derived from the same processing pipeline and therefore share identical parameters. For our analysis, parameters were chosen to accommodate one-second audio segments sampled at 16 kHz, providing a balance between time and frequency resolution.

- **Window function = Hanning**
  This parameter specifies the type of window function to apply to each segment of the signal during analysis. For our setup, we use Hanning windows, as they are well-suited for general-purpose applications and effectively reduce spectral leakage. In the future, we may experiment with other window types, such as Hamming, to observe how they impact performance and whether they offer any advantages for our specific use case.

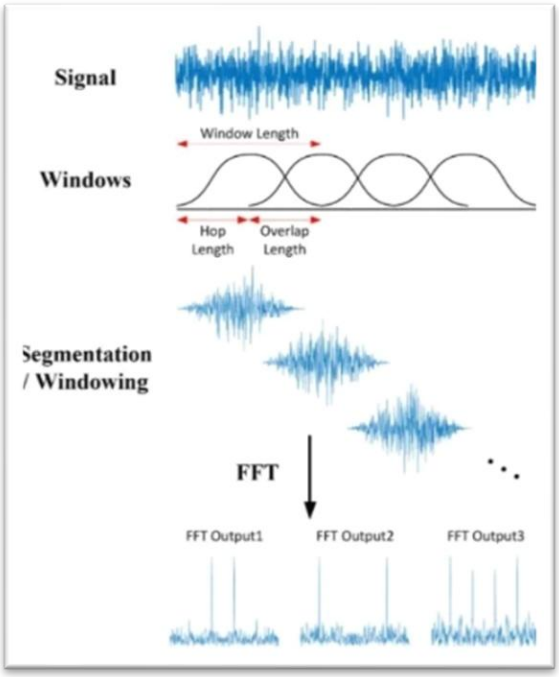


Figure 7.2.8

- **n-fft = 2048**
  This parameter determines the window length, where the value specifies the number of samples in each window. Our lower limit for this parameter is derived from the analysis in the previous chapter (7.2.6), which set a minimum window duration of 100 ms. At a sampling rate of 16,000 Hz, this translates to $16000 \cdot \frac{1000ms}{100ms} = 1600$ samples.
  However, it is common practice for this parameter to be a power of 2, as it facilitates efficient computation, particularly for operations like the Fast Fourier Transform (FFT). Therefore, the closest power of 2 is 2048, which we will use.

- **Hop length = 512**
  The hop length parameter determines the step size, in samples, between consecutive windows during the analysis. This value directly affects the overlap between windows, with smaller hop lengths resulting in greater overlap and finer time resolution. For our analysis, we chose a hop length of 512 samples. At a sampling rate of 16,000 Hz, this corresponds to a step size of $\frac{512}{16000} = 32ms$. Which results in an overlap of 75%.

- **n-mels = 128**
  This parameter determines the number of Mel filters into which the Mel filter bank will be divided. A larger number of Mel filters provides better resolution, allowing for finer frequency detail in the representation. However, the downside of using more Mel filters is that it generates more data, which increases computational requirements and can potentially slow down processing. We chose 128 because it is a common “sweet spot” that provides fine-grained detail without going overboard.

Parameters for the MFCC will be calculated in the next report as we are currently focusing on Mel-spectrogram.

7.2.9 Model Design

Machine Learning vs. Deep Learning

- **Traditional Machine Learning (ML):** Methods such as Support Vector Machines (SVM) or Random Forests require manually crafted features (e.g., statistical parameters, frequency bands). While they can be efficient with smaller datasets, they often struggle with high variability and noise.
- **Deep Learning (DL):** Neural networks (CNN, RNN, CRNN, etc.) learn feature representations directly from data. This can handle greater variability and noise in real-world scenarios (e.g., different drone models, flight conditions) but typically require larger datasets and more computational resources.

Image Processing Perspective

Even though the input data is audio, converting signals into a time-frequency representation (e.g., Mel spectrogram) creates a 2D "image." This approach allows:

- **Use of Convolutional Layers:** Convolutions excel at detecting local patterns or "shapes" in the spectrogram that correspond to drone rotor harmonics.
- **Easier Interpretation:** Spectrogram "images" are visually interpretable, letting us confirm that the model is focusing on relevant frequency bands and patterns.

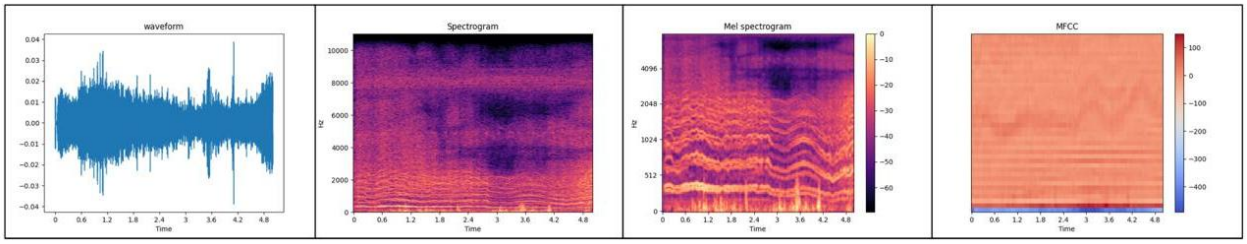


Figure 7.2.9.1

Why Choose CRNN?

Based on the article[96], several deep learning architectures were compared, including RNN, CNN, and CRNN. Their results (see attached table) demonstrate that while CNN models achieve slightly higher accuracy, they impose a significant increase in computational time. By contrast, CRNN offers a more balanced trade-off between accuracy and processing speed. Since we require real-time response on resource-limited hardware, CRNN becomes the most suitable choice.

Evaluation Metric	RNN	CNN	CRNN
CPU-Time (s)	333.45±60.90	957.33±320.01	487.53±178.75
Accuracy (%)	75.00±6.60	96.38±0.69	94.72±1.36
Precision (%)	75.92±10.30	96.24±0.81	95.02±1.14
Recall (%)	68.01±7.59	95.60±0.84	93.08±1.98
F1-score (%)	68.38±8.16	95.90±0.78	93.93±1.61

Figure 7.2.9.2

Training Time, Data Volume, and Resource Considerations

Deep learning requires a sufficiently large dataset to generalize well. CRNN, trained on diverse drone audio samples, can learn robust features with reasonable training resources. While a larger dataset can extend training time, the improved real-time performance justifies the additional upfront cost.

7.2.10 Summary

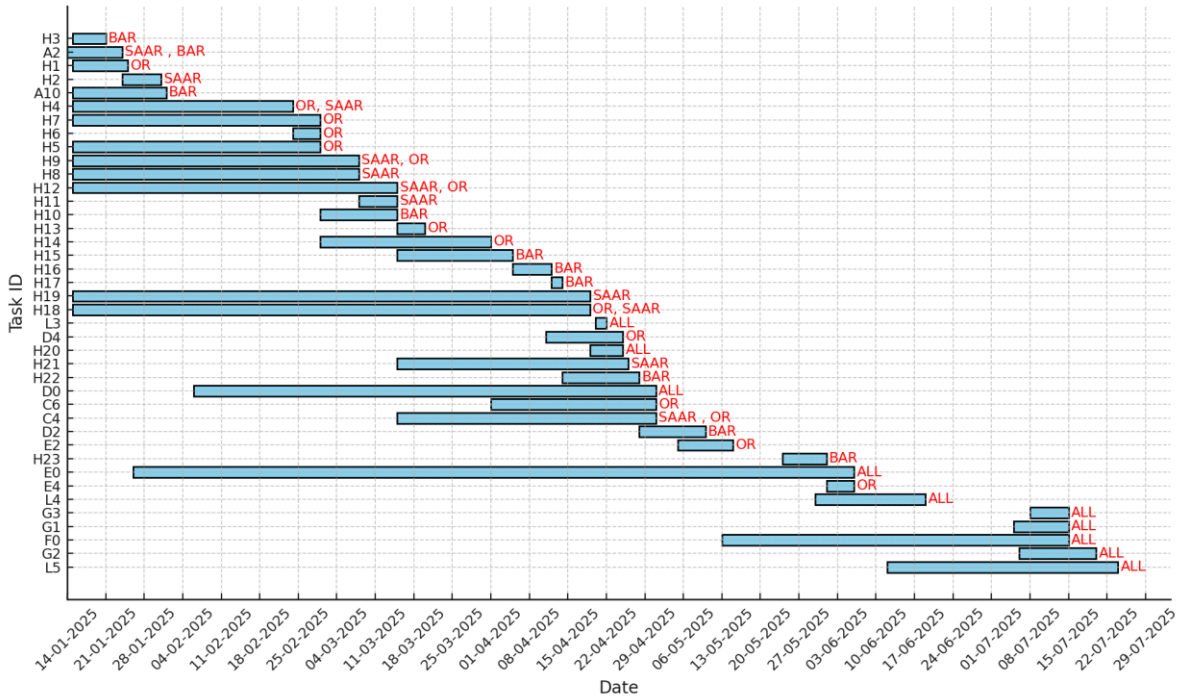
Alternative Microphone	Rode Video Micro
Microphone for Parabolic Reflector	Saramonic SR-LMX1+
Processor	Starting with Raspberry pi 4B and moving to Jetson nano if needed
Parabolic Reflector	Starting with diameter=0.33m, $\frac{f}{d} = 0.25$ with expected gain of 10db at minimum frequency. And developing a different design if needed
Dataset	Starting with a minor dataset from online sources. Planning to expand with self-made recordings.
Selected Feature	Starting with Mel-Spectrogram. Using also STFT, MFCC If needed
Feature Parameters	Starting with window: Hanning, window size(n_fft):2048, hop_length:512, n_mels:128. Exploring more parameters through the developing process.
Selected Model	Starting with custom CRNN. Exploring more if needed.

Table 7.2.10

7.3 Work Plan

7.3.1 Plan for Remaining Work

Task ID	Task Name	Start Date	End Date	Duration (days)	Dependencies	Responsibility	Assigned to
H3	Alternative Single Microphone testing	15.1.25	21.1.25	7		BAR	BAR
A2	Learning Drone Piloting	14.1.25	24.1.25	7		SAAR , BAR	SAAR , BAR
H1	Data generation for 0.1s model	15.1.25	25.1.25	1		SAAR	OR
H2	Saar Vacation	24.1.25	31.1.25	8		SAAR	SAAR
A10	Alternative Single Microphone Implementation.	15.1.25	1.2.25	14		BAR	BAR
H4	Data generation for multiclass model	15.1.25	24.2.25	3		SAAR	OR, SAAR
H5	0.1s model development	15.1.25	1.3.25	3	H1	OR	OR
H6	Multiclass model development	24.2.25	1.3.25	7	H4	OR	OR
H7	Bi-directional RNN model development	15.1.25	1.3.25	3		OR	OR
H8	General UAV's detection research	15.1.25	8.3.25	14		OR	SAAR
H9	General UAV's data collection	15.1.25	8.3.25	21		SAAR	SAAR, OR
H10	Localization approach research	1.3.25	15.3.25	14		BAR	BAR
H11	General UAV's data processing	8.3.25	15.3.25	2	H8,H9	SAAR	SAAR
H12	Expanded features research and testing	15.1.25	15.3.25	14		SAAR	SAAR, OR
H13	Expanded features data generation	15.3.25	20.3.25	1	H12	SAAR	OR
H14	GAN model development	1.3.25	1.4.25	14		OR	OR
H15	Choosing localization approach	15.3.25	5.4.25	14	H10	BAR	BAR
H16	Localization prototype for testing	5.4.25	12.4.25	7	H15	BAR	BAR
H17	Localization experiment	12.4.25	14.4.25	1	H16	BAR	BAR
H18	Recording setup	15.1.25	19.4.25	14		BAR	OR, SAAR
H19	Expanded dataset online research	15.1.25	19.4.25	3		SAAR	SAAR
L3	**Exhibition Poster**	**20.04.25**	**22.4.25**	**1**	**D4,H15**	**ALL**	**ALL**
H20	Self-Recorded Data Collection	19.4.25	25.4.25	2	H18	SAAR	ALL
D4	Final Software Design	11.04.25	25.4.25	10	H5,H6,H7,H14	OR	OR
H21	General UAV's algorithm development	15.3.25	26.4.25	7	H11	OR	SAAR
H22	Analyzing localization experiment results	14.4.25	28.4.25	14	H17	BAR, SAAR	BAR
C4	Real-Time Processing Development.	15.3.25	1.5.25	5	H12	SAAR , OR	SAAR , OR
C6	Decision-Making Algorithm Development.	1.4.25	1.5.25	7	D4	OR	OR
D0	**Final Design - Deadline**	**06.02.25**	**1.5.25**	**60**	**D4,H22**	**OR**	**ALL**
D2	Final Hardware Design Development.	28.4.25	10.5.25	10	H22	BAR	BAR
E2	Software Development and Testing.	05.05.25	15.5.25	7	C4	OR	OR
H23	Final localization approach implemintation	24.5.25	1.6.25	7	D2	BAR	BAR
E4	Hardware-Software Integration	1.6.25	6.6.25	6	H23,E2	OR	OR
E0	**Final Prototype - Deadline**	**26.01.25**	**6.6.25**	**90**	**E4**	**ALL**	ALL
L4	**Project Book Draft Submission**	**30.05.25**	**19.6.25**	**14**		**ALL**	ALL
F0	**Final Product - Deadline**	**13.05.25**	**15.7.25**	**45**	**G1,G3**	**ALL**	ALL
G1	Field Deployment for Testing and Documentation	05.07.25	15.7.25	7	E0	ALL	ALL
G3	Testing in the Exhibition Area	08.07.25	15.7.25	5	E0	ALL	ALL
G2	Documenting Results and Writing Detailed Report	06.07.25	20.7.25	10	F0	ALL	ALL
L5	**Final Project Delivery**	**12.06.25**	**24.7.25**	**30**	**G2**	**ALL**	ALL



7.3.2 Task Updates From SOW

Rescheduled Tasks:

Task ID	Task Name	Original Date	New Date	Delay (days)	Responsibility	Reason
E4	Hardware-Software Integration	5/25/2025	6/6/2025	12	OR	No Hardware Rady
E0	Final Prototype - Deadline	6/1/2025	6/6/2025	5	ALL	Need more time for Hardware devolepmant
D2	Final Hardware Design Development.	4/25/2025	5/10/2025	15	BAR	Unknown localization method.
C4	Real-Time Processing Development.	2/1/2025	5/1/2025	89	SAAR , OR	Unknown localization method.
C6	Decision-Making Algorithm Development.	2/15/2025	5/1/2025	75	OR	Unknown localization method.
A10	Alternative Single Microphone Implementatior	10/20/2024	2/1/2025	104	BAR	Microphone not working.
A2	Learning Drone Piloting	10/25/2024	1/24/2025	91	SAAR , BAR	

Delayed Tasks:

Task ID	Task Name	Original Date	Responsibili	Status	Reason
B1	Initial Microphone Array Design.	1.11.24	BAR	Undone	
B2	Initial Microphone Array Construction.	18.12.24	BAR	Undone	B1 undone
B3	Initial Microphone Array Testing.	20.12.24	BAR, SAAR	Undone	B2 undone
B8	Alternative Microphone Array Design.	1.12.24	BAR	Undone	
B9	Alternative Microphone Array Construction.	20.12.24	BAR	Undone	A10 undone
B10	Prototype 2 Construction.	25.12.24	ALL	Undone	B9 Undone
B11	Hardware-Software Integration	23.12.24	OR	Undone	B9 Undone
B0	**Prototype 2 - Deadline**	**1.1.25**	**ALL**	Partial	B9 Undone
C0	**Prototype 3 - Deadline**	**1.3.25**	**ALL**	**Canceled**	**delay with B0**

- Prototype 3 and all of its related tasks were cancelled due to the Delay of prototype 2.
- Prototype 3 is now planned to merge with the final product of the project.

8. Results

8.1 Data Collection and Dataset Summary

Data Sources

We collected audio data from various sources and conducted approximately 5,000 recordings to build a robust dataset. The data includes drone and background noise samples to ensure sufficient diversity for training and evaluation. Additionally, a substantial amount of background noise was collected to create a noise bank for data augmentation.

Original Dataset Composition

The following table summarizes the distribution of drone and non-drone samples across datasets:

Dataset Name	Drone Samples	Non-Drone Samples
Audio by Svanstrom	300	600
Binary_Drone_Audio	1,332	10,370
ESC-50	0	10,000
FSD-50K	0	95,384
MLP_2022_Synth_data	28,800	0
MLP_2022_real_data	6,194	0
Self-recorded	2,400	3,061

Table 8.1.1

Data Augmentation Process

As detailed in Section 7, extensive data augmentation was required to balance and enhance the dataset. The following table shows the augmentation percentages applied to each dataset:

Dataset Name	Shift	Gain	Time Stretch	Pitch	Noise	Augmented Needed	Final Count
Svanstrom	10%	20%	15%	20%	30%	600	900
Sara (Binary_Drone)	15%	20%	15%	15%	25%	1,268	2,600
Mic Array (Self-rec.)	10%	20%	10%	10%	20%	2,400	4,800
MLP_2022_real_data	10%	20%	10%	10%	25%	1,806	8,000
MLP_2022_Synth_data	-	20%	15%	15%	40%	1,400	5,000

Table 8.1.2

Final Dataset Composition

After augmentations and balancing, the final dataset is summarized in figure 8.1.3 as follows:

Label	Total Samples
Drone	60,300
Non-Drone	122,415

Table 8.1.3

Explanation

The data augmentation process involved shifting, scaling, pitch modulation, and adding noise to enhance robustness against real-world conditions. This step was essential to balance the dataset and increase model generalization.

8.2 Parabolic Microphone Development and Testing

As previously detailed in the planning stage, a parabolic reflector was designed.

Figure 8.2.1

Figure 8.2.2

The parts in figure 8.2.1 and 8.2.2 were 3D printed, and Saramonic SR-LMX1+ microphone was used. Rubber components were added at specific points to minimize noise and vibrations during operation. A series of experiments were conducted to evaluate the reflector's performance, focusing on key parameters such as frequency response, gain, and beamwidth.

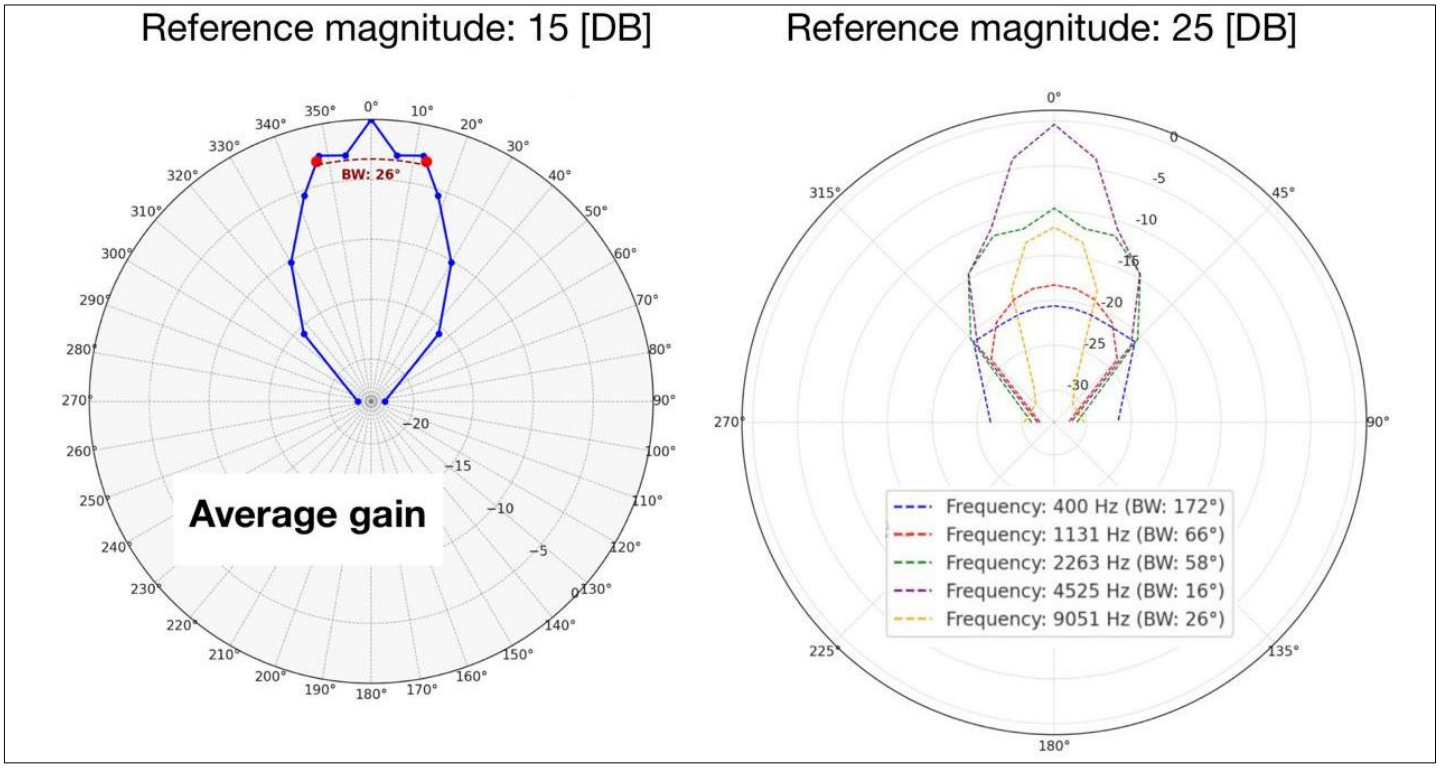


Figure 8.2.3

Figure 8.2.4

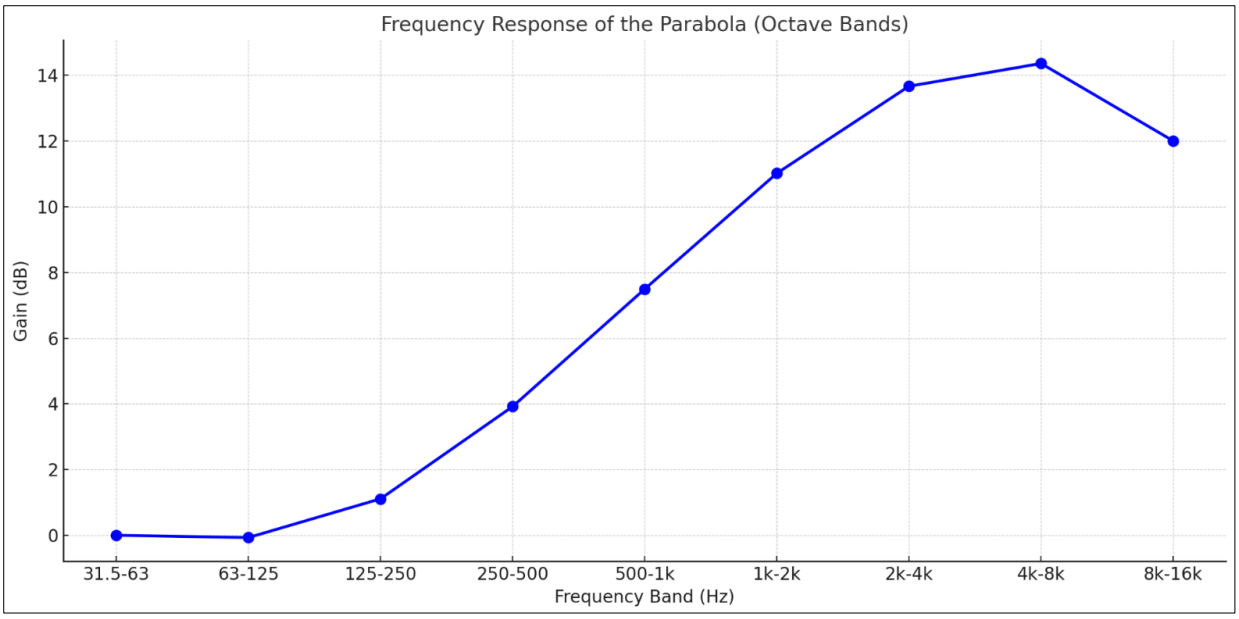


Figure 8.2.5

The experimental results are as follows:

- **Figure 8.2.4:** This polar plot shows the gain and beamwidth across different frequencies. The beamwidth varies between 16° and 26°, while the gain reaches up to 25 dB at certain frequencies.
- **Figure 8.2.5:** This graph illustrates the average frequency response of the reflector within the desired range of 1 kHz to 8 kHz. The reflector demonstrated an average gain of 15 dB and a peak gain of 25 dB in specific frequency bands.
- **Figure 8.2.3:** The normalized gain plot provides an averaged view of the reflector's performance over the target frequency range (1 kHz to 8 kHz). While the beamwidth is approximately 26°, the reflector maintains a gain of about 10 dB even when the beamwidth expands to around 60°, making it sufficient for the project's requirements.

The reflector's effectiveness was evident, as the drone sound was clearly audible at 100 meters, while without the reflector, it was barely noticeable at 20 meters.

Reflector Key Specifications

Plate Diameter	F/D Ratio	Beamwidth	Practical Beamwidth	Average Gain	Maximum Gain
0.33 meters	0.25	26°	60°	15 dB	25 dB

Table 8.2

The detailed experimental procedure and setup are documented in the appendix.

The next step involves fine-tuning the CRNN model to adapt it specifically to the frequency response and gain characteristics of the parabolic reflector, ensuring optimal alignment between the hardware and software components for drone detection tasks.

8.3 CRNN Model Performance and Metrics

As previously planned, the CRNN (Convolutional Recurrent Neural Network) model was developed to analyze Mel spectrograms for drone detection. The model was implemented using TensorFlow in a Linux environment running on Windows WSL. The architecture consists of convolutional layers for feature extraction, an LSTM layer for temporal pattern recognition, and dense layers for binary classification.

The model was trained on the datasets described in the data-collection section, excluding the MLSP_2022 dataset, which was unavailable at the time. In total, approximately 10,000 samples were used, divided into:

Dataset	percentage	Samples count
training set	70%	~6500
validation set	15%	~1400
test set	15%	~1400

Table 8.3.1

Model Architecture

- Convolutional Layers: Extract spatial features from the Mel spectrograms. The two convolutional layers have 64 and 128 filters, respectively, with kernel sizes of 3x3 and ReLU activations. MaxPooling layers follow each convolution to reduce spatial dimensions.

- LSTM Layer: Processes temporal dependencies in the spectrogram data, allowing the model to identify time-varying patterns characteristic of drones.

- Fully Connected Layers: These include a dense layer with 128 neurons and ReLU activation, followed by a final output layer with a sigmoid activation for binary classification.

- Dropout Layers: Used to reduce overfitting, with dropout rates of 0.3 after the LSTM layer.

During the training phase, the model was trained on a CPU i5 with 16 GB RAM. Each training session took several hours, which slowed down development and limited experimentation with larger model configurations.

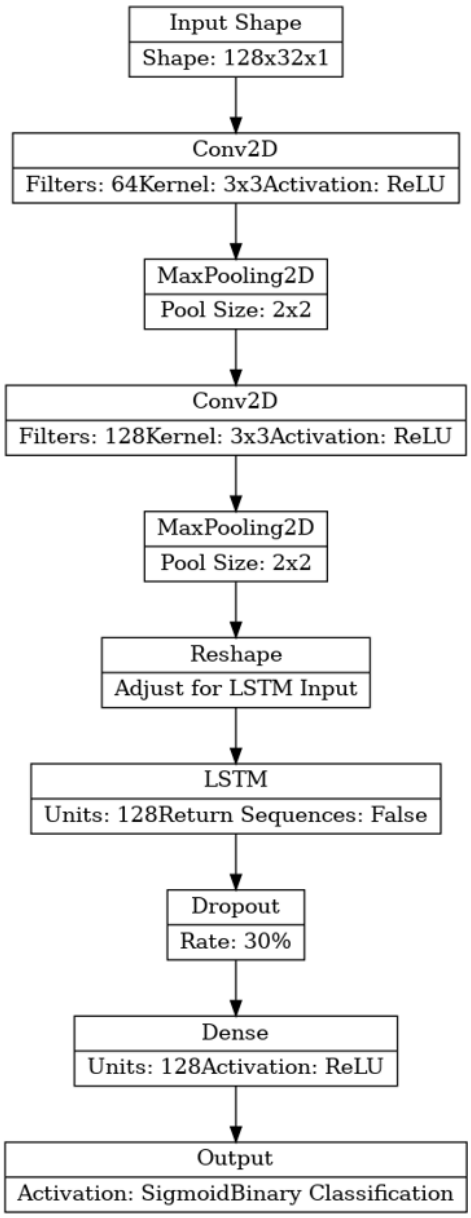


Figure 8.3

Model Results

Confusion Matrix

Actual \ Predicted	Non-drone	Drone
Non-drone	738 (TN)	43 (FP)
Drone	24 (FN)	588 (TP)

Table 8.3.2

Definition: A confusion matrix summarizes the model's classification performance. Rows represent the true class, and columns represent the predicted class.

Details:

- **True Negatives (TN):** 738 Non-drone samples were correctly classified.
- **False Positives (FP):** 43 Non-drone samples were misclassified as drones.
- **False Negatives (FN):** 24 Drone samples were misclassified as non-drones.
- **True Positives (TP):** 588 Drone samples were correctly classified.

Classification Report

	precision	recall	f1-score	support
0	0.97	0.94	0.96	781
1	0.93	0.96	0.95	612

Table 8.3.3

Details:

- **Precision:** Measures the proportion of true positives among all predicted positives. For drones, the precision is 93%, while for non-drones, it is 97%.
- **Recall:** Measures the proportion of true positives among all actual positives. The recall for drones is 96%, indicating a low rate of false negatives.
- **F1-Score:** The harmonic mean of precision and recall, providing a balanced measure. Both classes achieve high F1-scores of approximately 0.95–0.96.
- **support:** refers to the number of true instances for each class.
- **Accuracy:** The overall model accuracy is 95%, which is significantly higher than the 80% detection rate required by the project objectives.

Model Response Time

The system's maximum response time is 1.1 seconds, which is well below the project requirement of 5 seconds. This represents an improvement of 78% over the required response time.

Current Computing Power

With the addition of new hardware, we now have access to significantly enhanced computational resources:

- GPU: RTX 3090 GAMING PRO (24 GB)

- CPU: Intel i7 (14th generation)

- RAM: 32 GB

These resources enable faster training, allowing for larger-scale models and more extensive experimentation. This improvement resolves the limitations experienced during the initial development phase and opens the door to scaling the model for enhanced performance.

Future Plans

With improved computational resources now available, the following upgrades are planned:

- **Expand the Dataset:** Increase the size and diversity of the training data to further improve model generalization.
- **Enhance Model Parameters:** Increase the number of model parameters to allow for more complex pattern recognition.
- **Preprocessing Adjustments:** Train the model on spectrogram images with the bottom rows (frequencies below 1 kHz) removed, optimizing performance for the parabolic reflector.
- **Bidirectional RNN:** Experiment with a bidirectional RNN for better temporal context.
- **Advanced Features:** Explore heavier features such as STFT (Short-Time Fourier Transform) with higher resolution.
- **Model Deployment on Jetson Orin Nano:** Implement the trained model for real-time operations, ensuring efficiency and reliability in field environments.

8.4 Software Architecture and System Design

The system was developed in a Linux environment using Python as the primary programming language and VSCode as the integrated development environment (IDE). The architecture employs multithreading to ensure real-time operation and minimal delays. Locks and circular buffers are used to guarantee data reliability and prevent contention between threads. The system was designed using object-oriented programming (OOP) principles, which enhanced modularity and allowed for clear separation of responsibilities across the different components. This modular design, as illustrated in the block diagram, simplifies debugging, scalability, and future enhancements

Code diagram:

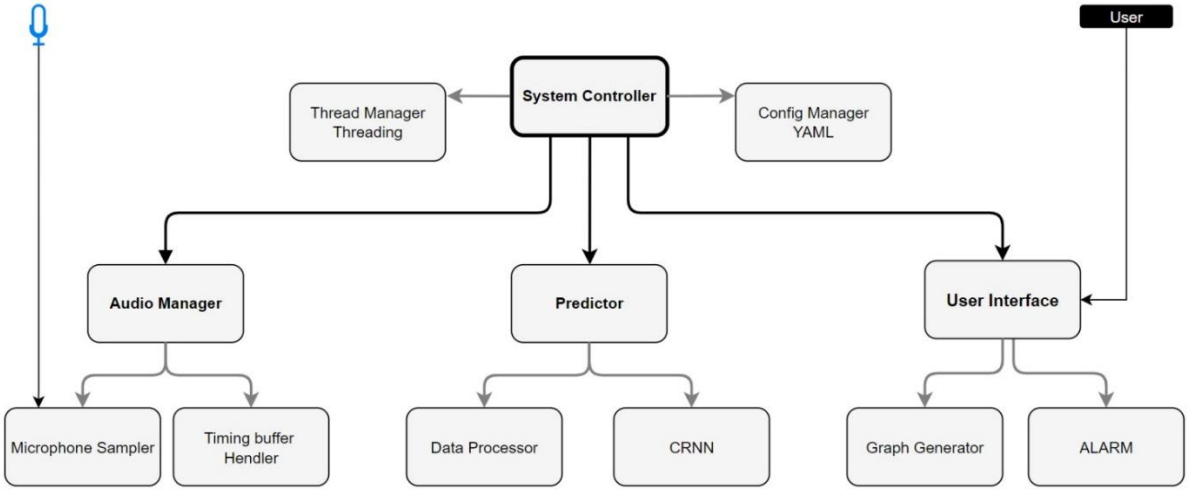


Figure 8.4.1

Block Descriptions:

1. System Controller

Functionality: Centralizes communication and coordination between all system modules.

Key Components:

- Thread Manager: Allocates threads for the Audio Manager, Predictor, and User Interface, ensuring concurrent operation without blocking.
- Config Manager (YAML): Loads and manages configuration files, including audio parameters, prediction thresholds, and system behavior settings.

2. Audio Manager

Functionality: Manages real-time audio data collection and preprocessing.

Key Components:

- Microphone Sampler: Collects raw audio signals from the microphone in real time.
- Timing Buffer Handler: Segments audio into desired sizes, manages overlaps, timestamps each sample, and updates the circular buffer for real-time access by the System Controller.

3. Predictor

Functionality: Processes audio data from the Timing Buffer and determines whether a drone is present.

Key Components:

- Data Processor: Extracts features (e.g., Mel spectrograms) from the segmented audio samples.
- CRNN: A trained Convolutional Recurrent Neural Network that classifies the processed features and predicts the presence of a drone.

4. User Interface

Functionality: Provides feedback to the user and facilitates interaction.

Key Components:

- Graph Generator: Creates real-time visualizations of audio spectrograms and system performance metrics.
- Alarm: Issues alerts (visual or auditory) upon detecting a drone.

Thread Management

The Audio Manager, Predictor, and User Interface operate on independent threads. The Timing Buffer Handler works alongside the Microphone Sampler within the Audio Manager to maintain continuous data flow.

8.5 Prototype Development

The current focus is on building the second prototype. The first prototype was successfully developed in alignment with the goals outlined in the SOW, serving as a foundational model for testing the system's core functionalities. Below are the key aspects of the second prototype in its current state:

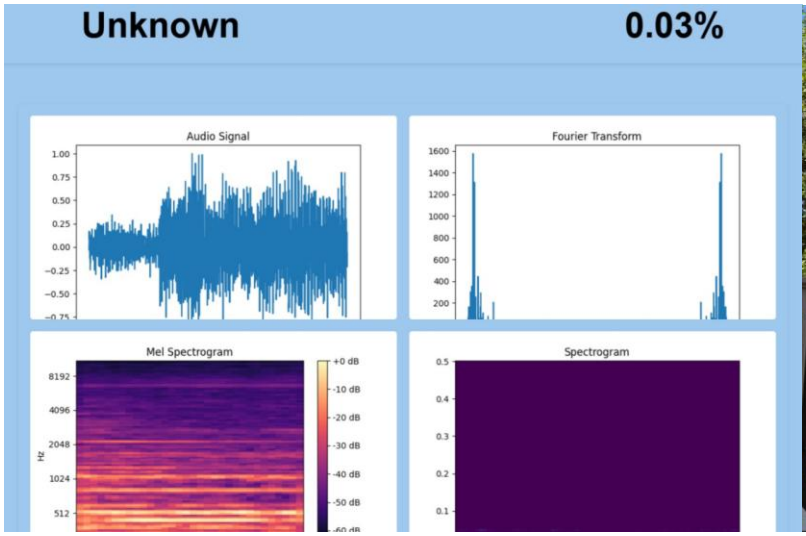


Figure 8.5.1

Figure 8.5.2

Design and Components.

The prototype consists of three parabolic antennas mounted on a tripod, as shown in Figure 8.5.2. Currently, the integration between the parabolic microphones has not yet been completed. The prototype operates wirelessly using a Raspberry Pi, which creates a Wi-Fi network for remote interaction and configuration. The system interface, shown in Figure 8.5.1, displays the model's prediction and the probability percentages of the decision at the top of the screen. The graphs at the bottom are temporary and are used for monitoring and controlling various stages within the system. The entire system is fully portable, weighing less than 7 kilograms, making it easy to transport between locations.

Power Supply

The prototype currently uses a portable power bank with a capacity of 3Ah and a power output of 65 W. This setup has demonstrated excellent performance so far, and we consider this solution for the final design.

Preliminary Testing

Initial tests with a basic microphone have shown outstanding results. However, detailed performance metrics are not presented at this stage, as professional testing has not yet been conducted.

Future Plans

- Integration with the parabolic microphones will be a key focus in the next phase of development.
- Once the directional capabilities of the parabolic microphones are fully developed and tested, they will be incorporated into the prototype.

8.6 Summary

Category	Details
Data Collection	Built a robust dataset with over 200,000 samples, including 60,300 drone samples and 122,415 background samples.
Model Development	CRNN achieved 95% accuracy with a response time of 1.1 seconds.
Parabolic Microphone	Beamwidth measured 26° on average with a practical beamwidth of 60°, achieving consistent gain across desired frequency bands.
Prototype Development	The first prototype was completed as per the SOW. The second prototype is under construction with upgrades, including Jetson Nano and parabolic microphone integration with directional capabilities.

Table 8.6.1

Progress Toward Goals

The current stage demonstrates significant progress toward meeting the project's objectives:

- Detection Metrics: The CRNN model achieved 95% accuracy, exceeding the defined target of 80%. The response time is 1.1 seconds, significantly better than the 5-second target.
- Directional Capabilities: While beamwidth measurements of 26° and practical coverage of 60° align with project requirements, further testing and integration of directional capabilities are planned for the next phases.
- Prototype Development: The first prototype fulfilled foundational requirements, and the second prototype builds upon this progress with enhanced processing and integration capabilities.

9. summary of Changes & Risks

9.1 summary of Changes

No significant changes have been made since the Project Charter. The project continues as planned.

9.2 Risks

9.2.1 Risk in Microphone Implementation

- **1. Parabolic Microphone**
  The developed parabolic microphone may fail to deliver adequate performance with the model, and there may not be enough time to develop an additional parabolic microphone.
- **2. Alternative Microphone**
  Difficulty in acquiring an alternative microphone due to budget overruns and uncertainty regarding the suitability of the current alternative microphone.

Implications:

- Inability to meet the directionality metric.
- Inability to meet the distance metric.

Alternative:

- The model will be tested using recordings made at specific distances with a smartphone.
- A risk remains in detecting quiet drones, as these microphones lack amplification and are not designed for long-range use.

9.2.2 Risk in Microphone Array Development

To ensure the microphone array functions properly, techniques such as beamforming or other methods are required to focus on a specific direction.

- **Microphone Array Development:**
  Developing a microphone array using either a parabolic reflector or shotgun microphones presents challenges such as synchronizing between microphones, calibrating sound levels and sampling times, and ensuring full area coverage. Failure to address these challenges may hinder the array's functionality.
- **Algorithmic Development:**
  Directional microphones solve the problem at long ranges, but algorithmic techniques are still required to handle close-range scenarios where sound overlaps across microphones.

Alternative:

- Train the model on a suitable dataset recorded from the array, enabling the model to independently identify patterns within the raw data from the array.

Dependency:

- Completion of the dataset at least two months before the project deadline.

Implications:

- Risk of failing to meet the directionality metric.

Alternative:

- Use a single microphone with a motorized mechanism to scan the area.

9.2.3 Risk in Developing a Model for UAVs (Non-Drone Aircraft)

Due to difficulties in obtaining data for UAVs, there is a significant risk that a model for these types of aircraft will not be developed.

Alternative:

- If time allows, an algorithmic approach will be tested on a small set of examples collected from media and online sources.
- Alternatively, no model will be developed for UAVs.

10. References

[1] Seidaliyeva U, Ilipbayeva L, Taissariyeva K, Smailov N, Matson ET. Advances and Challenges in Drone Detection and Classification Techniques: A State-of-the-Art Review. *Sensors*. 2024; 24(1):125. Retrieved from: https://www.mdpi.com/1424-8220/24/1/125

[2] Y. Wang, F. E. Fagian, K. E. Ho, and E. T. Matson, "A feature engineering focused system for acoustic uav payload detection," in 2022 Fourteenth International Conference on Agents and Artificial Intelligence (ICAART), ICAART, 2022.

[3] S. Latif, H. Cuayáhuitl, F. Pervez, F. Shamshad, H. S. Ali, and E. Cambria, "A survey on deep reinforcement learning for audio-based applications," arXiv preprint arXiv:2101.00240, 2021.

[4] G. Sharma, K. Umapathy, and S. Krishnan, "Trends in audio signal feature extraction methods," Applied Acoustics, vol. 158, p. 107 020, 2020.

[5] G. Nguyen, S. Dlugolinsky, M. Bobák, et al., "Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey," Artificial Intelligence Review, vol. 52, no. 1, pp. 77–124, 2019.

[6] Y. Wang, J. Wei-Kocsis, J. A. Springer, and E. T. Matson, "Deep learning in audio classification," in Information and Software Technologies: 28th International Conference, ICIST 2022, Kaunas, Lithuania, October 13–15, 2022, Proceedings, Springer, 2022, pp. 64–77.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25, 2012.

[8] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," Artificial intelligence review, vol. 53, no. 8, pp. 5455–5516, 2020.

[9] S. Latif, H. Cuayáhuitl, F. Pervez, F. Shamshad, H. S. Ali, and E. Cambria, "A survey on deep reinforcement learning for audio-based applications," arXiv preprint arXiv:2101.00240, 2021.

[10] E. Tzeng, J. Homan, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7167–7176.

[11] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," IEEE/ACM Transactions on audio, speech, and language processing, vol. 22, no. 10, pp. 1533–1545, 2014.

[12] M. Dong, “Convolutional neural network achieves human-level accuracy in music genre classification,” arXiv preprint arXiv:1802.09697, 2018.

[13] S. R. Park and J. Lee, “A fully convolutional neural network for speech enhancement,” arXiv preprint arXiv:1609.07132, 2016.

[14] Y. Chen, Q. Guo, X. Liang, J. Wang, and Y. Qian, “Environmental sound classification with dilated convolutions,” Applied Acoustics, vol. 148, pp. 123–132, 2019.

[15] Z. C. Lipton, J. Berkowitz, and C. Elkan, “A critical review of recurrent neural networks for sequence learning,” arXiv preprint arXiv:1506.00019, 2015.

[16] S. Latif, J. Qadir, A. Qayyum, M. Usama, and S. Younis, “Speech technology for healthcare: Opportunities, challenges, and state of the art,” IEEE Reviews in Biomedical Engineering, vol. 14, pp. 342–356, 2020.

[17] A. Sherstinsky, “Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network,” Physica D: Nonlinear Phenomena, vol. 404, p. 132 306, 2020.

[18] K. Cho, B. Van Merriënboer, C. Gulcehre, et al., “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” arXiv preprint arXiv:1406.1078, 2014.

[19] T. N. Sainath and B. Li, “Modeling time-frequency patterns with lstm vs. convolutional architectures for lvcsr tasks,” 2016.

[20] J. Li, A. Mohamed, G. Zweig, and Y. Gong, “Lstm time and frequency recurrence for automatic speech recognition,” in 2015 IEEE workshop on automatic speech recognition and understanding (ASRU), IEEE, 2015, pp. 187–191.

[21] D. Ghosal and M. H. Kolekar, “Music genre recognition using deep neural networks and transfer learning.,” in Interspeech, 2018, pp. 2087–2091.

[22] Y. Qian, M. Bi, T. Tan, and K. Yu, “Very deep convolutional neural networks for noise robust speech recognition,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, no. 12, pp. 2263–2276, 2016.

[23] T.-W. Sun, “End-to-end speech emotion recognition with gender information,” IEEE Access, vol. 8, pp. 152 423–152 438, 2020.

[24] T. Kala and T. Shinozaki, “Reinforcement learning of speech recognition system based on policy gradient and hypothesis selection,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 5759–5763.

[25] A. Tjandra, S. Sakti, and S. Nakamura, “Sequence-to-sequence asr optimization via reinforcement learning,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 5829–5833.

[26] H. Chung, H.-B. Jeon, and J. G. Park, “Semi-supervised training for sequence-tosequence speech recognition using reinforcement learning,” in 2020 International Joint Conference on Neural Networks (IJCNN), IEEE, 2020, pp. 1–6.

[27] J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, “Robust automatic speech recognition: A bridge to practical applications,” 2015.

[28] R. Fakoor, X. He, I. Tashev, and S. Zarar, “Reinforcement learning to adapt speech enhancement to instantaneous input signal quality,” arXiv preprint arXiv:1711.10791, 2017.
[29] N. Alamdari, E. Lobarinas, and N. Kehtarnavaz, “Personalization of hearing aid compression by human-in-the-loop deep reinforcement learning,” IEEE Access, vol. 8, pp. 203 503–203 515, 2020.

[30] N. Jaques, S. Gu, R. E. Turner, and D. Eck, “Generating music by fine-tuning recurrent neural networks with reinforcement learning,” 2016. 106

[31] N. Kotecha, “Bach2bach: Generating music using a deep reinforcement learning approach,” arXiv preprint arXiv:1812.01060, 2018.

[32] J. Xie and M. Zhu, “Handcrafted features and late fusion with deep learning for bird sound classification,” Ecological Informatics, vol. 52, pp. 74–81, 2019.

[33] J. Salamon and J. P. Bello, “Deep convolutional neural networks and data augmentation for environmental sound classification,” IEEE Signal processing letters, vol. 24, no. 3, pp. 279–283, 2017.

[34] J. Nam, K. Choi, J. Lee, S.-Y. Chou, and Y.-H. Yang, “Deep learning for audio-based music classification and tagging: Teaching computers to distinguish rock from bach,” IEEE signal processing magazine, vol. 36, no. 1, pp. 41–51, 2018.

[35] J. Abeer, “A review of deep learning based methods for acoustic scene classification,” Applied Sciences, vol. 10, no. 6, 2020.

[36] H. Seo, J. Park, and Y. Park, “Acoustic scene classification using various pre-processed features and convolutional neural networks,” in Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE), New York, NY, USA, 2019, pp. 25–26.

[37] V. Lostanlen, J. Salamon, M. Cartwright, et al., “Per-channel energy normalization: Why and how,” IEEE Signal Processing Letters, vol. 26, no. 1, pp. 39–43, 2018.

[38] Y. Wu and T. Lee, “Enhancing sound texture in cnn-based acoustic scene classification,” in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 815–819.

[39] G. Nguyen, S. Dlugolinsky, M. Bobák, et al., “Machine learning and deep learning frameworks and libraries for large-scale data mining: A survey,” Artificial Intelligence Review, vol. 52, no. 1, pp. 77–124, 2019.

[40] O. Mariotti, M. Cord, and O. Schwander, “Exploring deep vision models for acoustic scene classification,” Proc. DCASE, pp. 103–107, 2018.

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

[42] J. F. Gemmeke, D. P. Ellis, D. Freedman, et al., “Audio set: An ontology and humanlabeled dataset for audio events,” in 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, 2017, pp. 776–780. 105

[43] J. Salamon and J. P. Bello, “Deep convolutional neural networks and data augmentation for environmental sound classification,” IEEE Signal processing letters, vol. 24, no. 3, pp. 279–283, 2017.

[44] K. Koutini, H. Eghbal-zadeh, and G. Widmer, “Receptive-field-regularized cnn variants for acoustic scene classification,” arXiv preprint arXiv:1909.02859, 2019.

[45] D. S. Park, W. Chan, Y. Zhang, et al., “Specaugment: A simple data augmentation method for automatic speech recognition,” arXiv preprint arXiv:1904.08779, 2019.

[46] M. Lasseck, “Acoustic bird detection with deep convolutional neural networks,” in Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018), 2018, pp. 143–147.

[47] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., “Generative adversarial nets,” Advances in neural information processing systems, vol. 27, 2014.

[48] G. Sharma, K. Umapathy, and S. Krishnan, “Trends in audio signal feature extraction methods,” Applied Acoustics, vol. 158, p. 107 020, 2020.

[49] P. Podder, T. Z. Khan, M. H. Khan, and M. M. Rahman, “Comparative performance analysis of hamming, hanning and blackman window,” International Journal of Computer Applications, vol. 96, no. 18, 2014.

[50] C. Pan, “Gibbs phenomenon removal and digital filtering directly through the fast fourier transform,” IEEE Transactions on Signal Processing, vol. 49, no. 2, pp. 444–448, 2001.

[51] R. Bachu, S. Kopparthi, B. Adapa, and B. Barkana, “Separation of voiced and unvoiced using zero crossing rate and energy of the speech signal,” in American Society for Engineering Education (ASEE) zone conference proceedings, American Society for Engineering Education, 2008, pp. 1–7.

[52] B. Kedem, “Spectral analysis and discrimination by zero-crossings,” Proceedings of the IEEE, vol. 74, no. 11, pp. 1477–1493, 1986.

[53] J. Saunders, “Real-time discrimination of broadcast speech/music,” in 1996 IEEE international conference on acoustics, speech, and signal processing conference proceedings, IEEE, vol. 2, 1996, pp. 993–996.

[54] X. Yang, B. Tan, J. Ding, J. Zhang, and J. Gong, “Comparative study on voice activity detection algorithm,” in 2010 International Conference on Electrical and Control Engineering, 2010, pp. 599–602. doi:10.1109/iCECE.2010.153.

[55] Y. Korkmaz, A. Boyac, and T. Tuncer, “Turkish vowel classification based on acoustical and decompositional features optimized by genetic algorithm,” Applied Acoustics, vol. 154, pp. 28–35, 2019.

[56] D. Mitrovi, M. Zeppelzauer, and C. Breiteneder, “Features for content-based audio retrieval,” in Advances in computers, vol. 78, Elsevier, 2010, pp. 71–150.

[57] C. Saitis, K. Siedenburg, P. M. Schuladen, and C. Reuter, The role of attack transients in timbral brightness perception. Universitätsbibliothek der RWTH Aachen, 2019.

[58] X. Valero and F. Alas, “Applicability of mpeg-7 low level descriptors to environmental sound source recognition,” in Proceedings 1st Euroregio Conference, Ljubjana, 2010.

[59] G. Muhammad and K. Alghathbar, “Environment recognition from audio using mpeg-7 features,” in 2009 Fourth International Conference on Embedded and Multimedia Computing, IEEE, 2009, pp. 1–6.

[60] J. P. Teixeira and A. Gonçalves, “Algorithm for jitter and shimmer measurement in pathologic voices,” Procedia Computer Science, vol. 100, pp. 271–279, 2016.

[61] X. Li, J. Tao, M. T. Johnson, et al., “Stress and emotion classification using jitter and shimmer features,” in 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP’07, IEEE, vol. 4, 2007, pp. IV–1081.

[62] M. Farrús, J. Hernando, and P. Ejarque, “Jitter and shimmer measurements for speaker recognition,” in 8th Annual Conference of the International Speech Communication Association; 2007 Aug. 27-31; Antwerp (Belgium).[place unknown]: ISCA; 2007. p. 778-81., International Speech Communication Association (ISCA), 2007.

[63] K. Jensen, “Pitch independent prototyping of musical sounds,” in 1999 IEEE Third Workshop on Multimedia Signal Processing (Cat. No. 99TH8451), IEEE, 1999, pp. 215– 220.

[64] M. Jalil, F. A. Butt, and A. Malik, “Short-time energy, magnitude, zero crossing rate and autocorrelation measurement for discriminating voiced and unvoiced segments of speech signals,” in 2013 The international conference on technological advances in electrical, electronics and computer engineering (TAEECE), IEEE, 2013, pp. 208–212.

[65] V. Peltonen, J. Tuomi, A. Klapuri, J. Huopaniemi, and T. Sorsa, “Computational auditory scene recognition,” in 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, IEEE, vol. 2, 2002, pp. II–1941.

[66] A. Rabaoui, M. Davy, S. Rossignol, and N. Ellouze, “Using one-class svms and wavelets for audio surveillance,” IEEE Transactions on information forensics and security, vol. 3, no. 4, pp. 763–775, 2008.

[67] D. Smith, E. Cheng, and I. Burnett, “Musical onset detection using mpeg-7 audio descriptors,” in Proceedings of the 20th international congress on acoustics (ICA), Sydney, Australia, vol. 2327, 2010, p. 1014.

[68] Y. Ando, “Autocorrelation-based features for speech representation,” in Proceedings of Meetings on Acoustics ICA2013, Acoustical Society of America, vol. 19, 2013, p. 060 033.

[69] Z. Fu, G. Lu, K. M. Ting, and D. Zhang, “A survey of audio-based music classification and annotation,” IEEE transactions on multimedia, vol. 13, no. 2, pp. 303–319, 2010.

[70] J.-C. Wang, J.-F. Wang, K. W. He, and C.-S. Hsu, “Environmental sound classification using hybrid svm/knn classifier and mpeg-7 audio low-level descriptor,” in The 2006 IEEE international joint conference on neural network proceedings, IEEE, 2006, pp. 1731–1735.

[71] W. Yang and S. Krishnan, “Combining temporal features by local binary pattern for acoustic scene classification,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 25, no. 6, pp. 1315–1321, 2017.

[72] X. Valero and F. Alas, “Classification of audio scenes using narrow-band autocorrelation features,” in 2012 Proceedings of the 20th European signal processing conference (EUSIPCO), IEEE, 2012.

[73] G. Percival and G. Tzanetakis, “Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses,” IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 22, no. 12, pp. 1765–1776, 2014.

[74] J. Bispham, “Rhythm in music: What is it? who has it? and why?” Music perception, vol. 24, no. 2, pp. 125–134, 2006.

[75] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” IEEE Transactions on speech and audio processing, vol. 10, no. 5, pp. 293–302, 2002. 109

[76] D. Sztahó, M. G. Tulics, K. Vicsi, and I. Valálik, “Automatic estimation of severity of parkinsons disease based on speech rhythm related features,” in Proceedings of 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom 2017) Debrecen, Hungary, 2017, pp. 11–16.

[77] B. Ulriksson, “Conversion of frequency-domain data to the time domain,” Proceedings of the IEEE, vol. 74, no. 1, pp. 74–77, 1986.

[78] L. Lee and W. E. L. Grimson, “Gait analysis for recognition and classification,” in Proceedings of Fifth IEEE International Conference on Automatic Face Gesture Recognition, IEEE, 2002, pp. 155–162.

[79] M. A. Bartsch and G. H. Wakefield, “Audio thumbnailing of popular music using chroma-based representations,” IEEE Transactions on multimedia, vol. 7, no. 1, pp. 96– 104, 2005.

[80] T. Li, M. Ogihara, and Q. Li, “A comparative study on content-based music genre classification,” in Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, 2003, pp. 282–289.

[81] L. Lu, D. Liu, and H.-J. Zhang, “Automatic mood detection and tracking of music audio signals,” IEEE Transactions on audio, speech, and language processing, vol. 14, no. 1, pp. 5–18, 2005.

[82] **"Classify Respiratory Abnormality in Lung Sounds Using STFT and a Fine-Tuned ResNet18 Network"**
https://arxiv.org/abs/2208.13943

[83] "Comparison of Time-Frequency Representations for Environmental Sound Classification using Convolutional Neural Networks"
https://arxiv.org/abs/1706.07156

[84] “Audio Recognition using Mel Spectrograms and Convolution Neural Networks”
https://noiselab.ucsd.edu/ECE228_2019/Reports/Report38.pdf

[85] “Guide to Audio Classification Using Deep Learning”
https://www.analyticsvidhya.com/blog/2022/04/guide-to-audio-classification-using-deep-learning/

[86] “Introduction to audio data - Hugging Face Audio Course”
https://huggingface.co/learn/audio-course/en/chapter1/audio_data

[87] mlearnere, Learning from audio: The mel scale, mel spectrograms, and mel frequency cepstral coecients, https://towardsdatascience.com/learning-from-audio-the-melscale-mel-spectrograms-and-mel-frequency-cepstral-coefficients-f5752b6324a8, 2022.

[88] P. Somervuo, A. Harma, and S. Fagerlund, “Parametric representations of bird sounds for automatic species recognition,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 14, no. 6, pp. 2252–2263, 2006.

[89] S. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences,” IEEE transactions on acoustics, speech, and signal processing, vol. 28, no. 4, pp. 357–366, 1980.

[90] A. Krueger and R. Haeb-Umbach, “Model-based feature enhancement for reverberant speech recognition,” IEEE Transactions on Audio, Speech, and Language Processing, vol. 18, no. 7, pp. 1692–1707, 2010.

[91] M. Sahidullah and G. Saha, “Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition,” Speech communication, vol. 54, no. 4, pp. 543–565, 2012.

[92] G. Kour and N. Mehan, “Music genre classification using mfcc, svm and bpnn,” International Journal of Computer Applications, vol. 112, no. 6, 2015.

[93] D. Sztahó, M. G. Tulics, K. Vicsi, and I. Valálik, “Automatic estimation of severity of Parkinson's disease based on speech rhythm related features,” in Proceedings of 8th IEEE International Conference on Cognitive Infocommunications, 2017.

[94] Jekateryńczuk, G.; Piotrowski, Z. A Survey of Sound Source Localization and Detection Methods and Their Applications. *Sensors* **2024**, *24*, 68. https://doi.org/10.3390/s24010068 https://www.mdpi.com/1424-8220/24/1/68

[95] Chen, T.; Yu, J.; Yang, Z. Research on a Sound Source Localization Method for UAV Detection Based on Improved Empirical Mode Decomposition. *Sensors* **2024**, *24*, 2701. https://doi.org/10.3390/s24092701 https://www.mdpi.com/1424-8220/24/9/2701

[96]“Drones Detection and Identification Using Deep Learning Techniques” by Sara Abdulrazaq Al-Emad.