A LARGE-SCALE UAV AUDIO DATASET AND AUDIO-BASED UAV CLASSIFICATION USING CNN

by

Yaqin Wang

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Polytechnic Institute

West Lafayette, Indiana

August 2023

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Eric T Matson, Chair

Department of Computer and Information Technology

Dr. Jin Wei-Kocsis

Department of Computer and Information Technology

Dr. John A Springer

Department of Computer and Information Technology

Dr. Piotr Artiemjew

Department of Mathematics and Computer Science, University of Warmia and Mazury

Dedicated to my grandma.

ACKNOWLEDGMENTS

I wish to thank my dissertation committee, without their guidance and mentoring, I would not have made it. Dr. Jin Wei-Kocsis, Dr. John A Springer, and Dr. Piotr Artiemjew served as supportive committee members, and Dr. Eric T Matson, my Chair, went above and beyond to help me reach my goal.

TABLE OF CONTENTS

LIST OF TABLES 8

LIST OF FIGURES 9

ABSTRACT 11

1 INTRODUCTION 12

1.1 Background 12

1.2 Research Questions 14

1.3 Significance 15

1.4 Limitations and Future Works 15

1.5 Summary 17

2 REVIEW OF LITERATURE 18

2.1 Background of Machine Learning and Deep Learning 18

2.1.1 Machine Learning in General 18

2.1.2 Deep Learning Models in Audio-based Applications 19

2.2 DRL Algorithms for Audio Classification 22

2.2.1 Model-based DRL 22

2.2.2 Policy Gradient-based DRL 23

2.2.3 Value-based DRL 24

2.3 Audio Data Pre-Processing and Augmentation 24

2.3.1 Data Pre-Processing 24

2.3.2 Data Augmentation 25

2.4 Recent Research on DL-based Audio Application 26

2.5 Evolution of Audio Features 28

2.6 Feature Extraction Methods for Audio 29

2.6.1 Window Function 29

2.6.2 Time Domain Features 31

Zero-crossing rate (ZCR) 31

Amplitude Descriptor (AD) . . . 32
Attach Delay Sustain Release (ADSR) Envelop . . . 32
Log Attack Time (LAT) . . . 32
Shimmer . . . 32
Short Time Energy (STE) . . . 33
Volume . . . 33
Temporal Centroid (TC) . . . 33
Auto-correlation Based Features . . . 33
Rhythm-based Features . . . 33
2.6.3 Frequency Domain Features . . . 34
Linear Predictive Coding (LPC) Coefficients . . . 34
Code Excited Linear Prediction (CELP) . . . 35
Linear Spectral Frequency . . . 35
Peak Frequency . . . 35
Spectrum Envelope . . . 36
Chroma Based Features . . . 36
Spectral Centroid . . . 36
2.6.4 Cepstral Domain Features . . . 38
Mel Frequency Cepstral Coefficients (MFCCs) . . . 38
Linear Prediction Cepstral Coefficients (LPCCs) . . . 38
Perceptual Linear Prediction (PLP) Cepstral Coefficients . . . 39
Relative-spectral PLP (RASTA-PLP) feature . . . 39
Greenwood function cepstral coefficients (GFCC) . . . 39
Gammatone cepstral coefficients (GTCCs) . . . 40
2.7 Feature Extraction Tools . . . 40
2.8 ML-Based UAV Classification with Computer Vision . . . 41
2.9 ML-Based UAV Classification with Radar . . . 45
2.10 ML-Based UAV Classification with Radio-Frequency . . . 48
2.11 ML-Based UAV Classification with Audio . . . 51
2.12 ML-based UAV Payload Detection . . . 53

2.13 Summary 55

3 METHODOLOGY 56

3.1 Data Collection 56

3.2 UAVs Specifications 57

3.3 Overall System 69

3.4 Audio Feature Extraction 70

3.5 Choice of Neural Network 72

3.6 Summary 75

4 EXPERIMENT 76

4.1 First Phase of the Experiment 76

4.1.1 UAV Audio Dataset 76

4.1.2 Data Collection 76

4.1.3 Dataset Evaluation for 10 Classes 77

4.1.4 Convolutional Neural Network Training 78

4.1.5 Results Evaluation and Analysis 79

4.2 Second Phase of the Experiment 81

4.2.1 Data Collection 81

4.2.2 Dataset Evaluation 81

4.2.3 Convolutional Neural Network Training 82

4.2.4 Result Evaluation and Analysis 83

4.3 Third Phase of the Experiment 87

4.3.1 Data Collection 87

4.3.2 Dataset Evaluation for 22 Classes 87

4.3.3 Convolutional Nerual Network Training 87

4.3.4 Result Evaluation and Analysis 91

4.4 Summary 94

5 CONCLUSION 96

REFERENCES 101

LIST OF TABLES

1.1 Solutions for UAV detection using vision, radar, radio-frequency, and acoustic . 13
2.1 Time Domain Features 30
2.2 Frequency Domain Features 34
2.3 Cepstral Domain Features 37
2.4 Research Papers on ML-based UAV detection with Computer Vision 42
2.5 Research Papers on ML-based UAV detection with Radar 44
2.6 Research Papers on ML-based UAV detection with Radio Frequency 47
2.7 Research Papers on ML-based UAV detection with Audio 50
2.8 Research Papers on ML-based UAV payload detection 53
3.1 UAVs Included in the Dataset 57
4.1 UAV Audio Dataset 10 Classes 77
4.2 Benchmark Evaluation Results with ML Models for 10 Classes 77
4.3 Accuracy, Precision, Recall, and F1-scores for 10 Classes Dataset 79
4.4 UAV Audio Dataset 15 Classes 80
4.5 Benchmark Evaluation Results with ML Models for 15 Classes 81
4.6 Accuracy, Precision, Recall, and F1-scores for 15 Classes Dataset 85
4.7 UAV Audio Dataset 22 Classes 86
4.8 Benchmark Evaluation Results with ML Models for 22 Classes 88
4.9 Accuracy, Precision, Recall, and F1-scores for 22 Classes Dataset 91
4.10 UAV Models and Labels 95

LIST OF FIGURES

1.1 ML structure for audio 14
2.1 Different structures of deep learning models[6] 20
2.2 Different DRL Structures [6] 23
2.3 Evolution of Audio Features 29
3.1 Outdoor Data Collection Site 56
3.2 Self-build David Tricopter 58
3.3 Self-build PhenoBee 59
3.4 Autel Evo 2 59
3.5 Yuneec Typhoon H Plus 60
3.6 Swellpro Splash 3 Plus 60
3.7 DJI Matrice 200 61
3.8 DJI Matrice 200 V2 61
3.9 DJI Mavic Air 2 62
3.10 DJI Mavic Mini 62
3.11 DJI Mini 2 63
3.12 DJI Mavic 2 Pro 63
3.13 DJI Air 2s 64
3.14 DJI Phantom 2 64
3.15 DJI Phantom 4 65
3.16 DJI RoboMaster TT Tello Talent 65
3.17 Hasakee Q11 66
3.18 Syma X5SW 66
3.19 Syma X5UW 67
3.20 Syma X20 67
3.21 Syma X20P 68
3.22 Syma X26 68
3.23 UDI U46 69
3.24 Audio-based UAV Classification System Overview 70

3.25 MFCC Features Extraction Process. 72
3.26 CNN Structure for 10 Classes 74
4.1 Evaluation Results for 10 Classes 78
4.2 CNN Structure for 15 Classes 82
4.3 Evaluation Results for 15 Classes 84
4.4 CNN Structure for 22 Classes 88
4.5 Evaluation Results for 22 Classes 90
4.6 Accuracy Score for Each Class 90
4.7 Confusion Matrix for 22 Classes 93
4.8 OvR ROC Curve 94

ABSTRACT

The growing popularity and increased accessibility of unmanned aerial vehicles (UAVs) have raised concerns about potential threats they may pose. In response, researchers have devoted significant efforts to developing UAV detection and classification systems, utilizing diverse methodologies such as computer vision, radar, radio frequency, and audio-based approaches. However, the availability of publicly accessible UAV audio datasets remains limited. Consequently, this research endeavor was undertaken to address this gap by undertaking the collection of a comprehensive UAV audio dataset, alongside the development of a precise and efficient audio-based UAV classification system.

This research project is structured into three distinct phases, each serving a unique purpose in data collection and training the proposed UAV classifier. These phases encompass data collection, dataset evaluation, the implementation of a proposed convolutional neural network, training procedures, as well as an in-depth analysis and evaluation of the obtained results. To assess the effectiveness of the model, several evaluation metrics are employed, including training accuracy, loss rate, the confusion matrix, and ROC curves.

The findings from this study conclusively demonstrate that the proposed CNN classifier exhibits nearly flawless performance in accurately classifying UAVs across 22 distinct categories.

1. INTRODUCTION

1.1 Background

Unmanned Air Vehicles (UAVs), commonly referred to as drones, have gained significant popularity in recent years, finding applications in both amateur sports events and homeland security. However, this surge in popularity has also given rise to a number of issues, including concerns surrounding privacy, as well as threats to airspace security and safety[1]. Despite the diverse intentions behind UAV usage, the potential threats they pose can result in significant damage, whether due to human error or intentional malicious activities. Disturbing incidents involving UAV threats have been reported worldwide, underscoring the seriousness of the issue. For instance, on January 18th, 2023, a deadly drone attack occurred in the heart of Abu Dhabi, claiming the lives of three individuals [2]. Furthermore, in 2019, a series of drone harassment incidents took place outside a firehouse in Salem, Virginia, which was located in close proximity to a memorial composed of steel beams retrieved from the wreckage of the North Tower of the World Trade Center, destroyed in the 9/11 terrorist attacks[3]. The harassment persisted and extended to the first responders' garage, intensifying the concerns surrounding these incidents.

Air control authorities worldwide are actively engaged in mitigating and potentially eradicating the risks associated with unmanned aerial vehicles (UAVs). However, it is crucial to acknowledge that regulatory frameworks and guidelines alone may not suffice in preventing intentional and criminal attacks. In addressing this concern, the development and implementation of cutting-edge technologies for the detection and classification of UAVs offer efficient solutions [4].

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 [5]. 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

Table 1.1. Solutions for UAV detection using vision, radar, radio-frequency, and acoustic

Method	Applications	Advantages	Disadvantages
Vision	Object detection, autonomous driving, facial recognition, object detection	Popular in recent years	Lack of publicly available datasets, Noise in visual data
Radar	Aviation and maritime traffic	Traditional, well-studied	Small radar cross-section, Low flying altitude
Radio frequency	Industrial, scientific, medical radio band (ISM band) at 2.4 GHz	Works in day and night, low cost, large detection range	Lack of RF signature database Frequency hopping
Acoustic	Smart microphones, Shazam app	Works day and night, low cost	Lack of robust datasets Background noise

yielding promising outcomes. Nonetheless, it is vital to recognize that each approach possesses its own set of strengths and limitations, which are succinctly summarized in Table **??**.

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 [6]. 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 1.1 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[7]. Initially, the system receives raw audio data samples as input, which then undergo 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 1.1. 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 [8]. 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.

1.2 Research Questions

The goal of this paper is to find solutions or answers to the following research questions.

- What are the current solutions for UAV detection and classification systems? What are the advantages and disadvantages of each approach?
- How well does the proposed solution perform for the audio-based UAV classification system?
- What are the limitations of the audio-based UAV classification system?

1.3 Significance

This paper makes a significant contribution in three key aspects. Firstly, it presents a comprehensive literature review of machine learning (ML) methodologies and architectures employed in audio-based UAV detection and classification systems. This review provides an extensive understanding of the existing knowledge in the field.

Secondly, a benchmark UAV audio dataset comprising 22 distinct types of UAVs is meticulously collected. This dataset serves as a valuable resource for future research and benchmarking purposes, facilitating advancements in the field of UAV audio classification.

Thirdly, a state-of-the-art UAV audio classifier is developed and trained using the aforementioned dataset. The classifier is based on the proposed methodology and exhibits impressive performance. The evaluation of the convolutional neural network (CNN) model showcases promising results, further validating its efficacy in accurately classifying UAV audio samples.

Overall, this paper's contributions lie in its comprehensive literature review, the creation of a benchmark UAV audio dataset, and the successful training and evaluation of a CNN-based UAV audio classifier, which collectively pave the way for advancements in the field of audio-based UAV detection and classification systems.

1.4 Limitations and Future Works

The limitations of this research can be summarized as follows:

- The dataset used in this study only includes Class I UAVs. Other classes of UAVs are not represented in the dataset, which may restrict the generalizability of the findings.

- Data collection for the dataset was limited to the period between sunrise and sunset. This restriction was imposed in accordance with regulations and for reasons of personal safety. Consequently, the dataset may not capture the full range of environmental conditions in which UAVs operate.

- The outdoor data collected for the dataset incorporates various noises inherent to countryside environments, including air traffic, ground traffic, birds, insects, wind, human conversation, and other factors. These ambient noises can introduce variability and potential interference in the audio recordings.

- The dataset encompasses 22 classes, each representing a distinct UAV model. Although this collection is currently one of the largest audio datasets available for UAVs, it is important to note that it does not encompass all possible UAVs from different manufacturers and models, indicating the potential for further dataset expansion.

- Each class within the dataset consists of a variable number of data entries, ranging from 100 to 138, with each entry being 5 seconds long. While this provides a foundation for analysis, the inclusion of a greater number of data entries per class would contribute to the robustness of the findings.

- The proposed classifier is based on a convolutional neural network (CNN) structure. This choice of architecture forms the basis of the classification system and may influence the performance and outcomes of the model.

All the aforementioned limitations represent potential avenues for future research. An essential objective of subsequent studies involves the ongoing expansion of the dataset to encompass a broader range of UAV types, as well as an increased volume of data for each class. Another significant goal is the incorporation of additional deep learning architectures, such as self-supervised learning and semi-supervised learning, leveraging the extensive dataset that has been amassed. Furthermore, conducting a comparative analysis of the performance achieved by different model structures would yield valuable insights into the classification of UAVs.

1.5 Summary

This chapter provides an extensive overview of the background information (Section 1.1) pertaining to recent unmanned aerial vehicle (UAV) threats and their potential implications. It also delves into the research questions (Section 1.2) that guide this study, elucidating the fundamental inquiries that underpin the investigation. Additionally, the chapter discusses the significance of this research (Section 1.3), highlighting its relevance and contributions to the field. Furthermore, it outlines the limitations encountered during the course of the study and offers insights into potential avenues for future research (Section 1.4).

2. REVIEW OF LITERATURE

2.1 Background of Machine Learning and Deep Learning

2.1.1 Machine Learning in General

Data needs to be prepared and preprocessed before the training phase. Various methods can be employed for data preparation, including feature engineering, exploratory data analysis (EDA), and data transformation. These methods can be further classified into subsets. Feature engineering encompasses feature selection, feature extraction methods, feature transformation, and more. EDA includes dimensional reduction, linear methods, and other techniques. Data transformation involves normalization, integration, and other procedures [8][9]. To address the issue of overfitting, several algorithms can be applied, such as the addition of dropout layers, pruning, and regularization techniques [6][10].

According to the assigned task, machine learning (ML) algorithms can be categorized into three groups: supervised learning, unsupervised learning, and reinforcement learning [11]. Supervised learning refers to a specific type of machine learning that utilizes labeled datasets to train models for solving classification or prediction problems [12]. 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 [13]. 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 [14]. 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 [15].

2.1.2 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), Sequence-to-Sequence (Seq2Seq) models, and Generative Models [16]. Figure 2.1 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 [16]. 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. Seq2Seq models, built upon the foundation of RNNs, are employed for tasks such as speech recognition, machine translation, and voice conversion. These models excel at mapping input sequences to output sequences, making them well-suited for tasks requiring sequential information processing. 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 audio-based 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

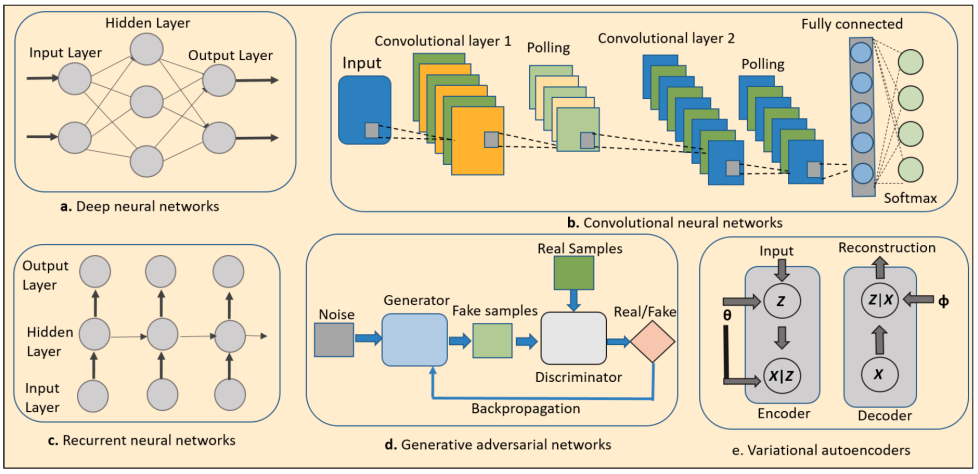


Figure 2.1. Different structures of deep learning models[6]

topologies, such as images [17]. 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 [18].

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, fully-convolutional 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 [19].

CNN models have found applications in various audio processing tasks, such as automatic speech recognition (ASR) [20], music genre classification [21], and speech enhancement [22]. Nevertheless, when it comes to processing raw audio waveforms with high sample rates, the limited receptive fields of CNNs can present challenges [6]. 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 [23].

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 offer 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 [24]. 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 [6]. Compared to the traditional Hidden Markov model (HMM) models, RNNs have produced better results in many audio and speech processing applications [25]. Because of these characteristics, two of the most popular RNN structures, Long-Short Term Memory (LSTM) [26] 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 [27]. In recent years, Time-Frequency LSTMs [28] and Frequency-LSTMs [29] 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 [25]. CRNN has been used in music classification[30], ASR [31], Speech Emotion Recognition (SER) [32], and more.

Sequence-to-sequence (Seq2Seq) models were created to solve problems with sequences of unspecified length [33]. They were used in machine translation first and later applied to many different applications with sequence modeling tasks. A Seq2Seq model can be divided into two parts: decoder and encoder. The encoder is one RNN that generates a vector representation from the input, and the decoder is another RNN that generates output by inheriting those learned features from the encoder. The structures of the Seq2Seq model can be unidirectional or bidirectional, single-layer or multi-layer [34]. Seq2Seq models have become more popular in audio and speech processing for their ability to convert the input to output sequences [6]. Different Seq2Seq models have been created and investigated by research in audio, speech, and language processing topics, including Recurrent Neural Aligner [35], Recurrent Neural Network Transducer [36], Transformer Networks [37], and more.

The three most used types of Generative Models are Generative Adversarial Networks (GANs) [38], autoregressive models [39], and Variational Autoencoders (VAEs) [40]. These models can read and learn the fundamental distributions in the speech dataset and have been extensively studied and used by audio and speech processing researchers. GANs and VAEs have been widely used in synthesizing speech. They have also been used to create more training data by generating features or speech data [6]. In the autoregressive models, future behavior is generated iteratively based on past behavior by using RNNs, such as LSTM or GRU structures.

2.2 DRL Algorithms for Audio Classification

Reinforcement Learning (RL) is a well-established branch of Machine Learning (ML) in which intelligent agents learn to take actions based on trial and error [41]. When RL is combined with Deep Learning (DL), it gives rise to Deep Reinforcement Learning (DRL). DRL has proven effective in handling more complex environments with high computational requirements or large state spaces. DRL utilizes Deep Neural Networks (DNNs) to evaluate models, policies, or values [16]. DRL can be categorized in various ways, such as policy-based versus value-based, model-free versus model-based, or on-policy versus off-policy DRL models. Figure 2.2 illustrates the different types of DRL models along with their main characteristics.

In the following paragraphs, we will delve into popular DRL models used in audio-based systems and discuss their applications in three sub-categories: model-based DRL, policy gradient-based DRL, and value-based DRL.

2.2.1 Model-based DRL

Model-based DRL algorithms depend on the environment, such as reward functions, along with a planning algorithm. Model-free DRL algorithms usually require a large amount of sample data to achieve acceptable results. Differently, model-based algorithms tend to produce results with improved sample and time efficiency [42]. Simulated policy learning (SimPLe) was proposed by [43]. It is a model-based DRL algorithm for video prediction. Fewer interactions of agent-environment are needed for SimPle than model-free algorithms.

Results show that SimPLe performs better than the state-of-the-art model-free algorithms in the games of Atari. Another model-based DRL algorithm was proposed by [44] called TreeQN. It was designed for a more complicated environment without the presence of the transition model. In TreeQN algorithm, Q-values are estimated by combining model-free and model-based methods.

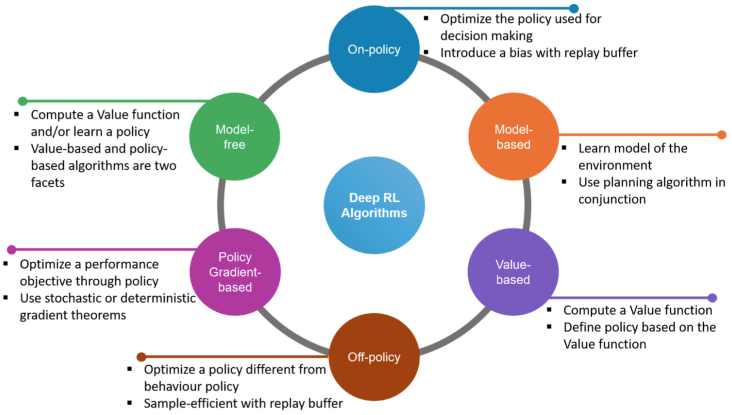


Figure 2.2. Different DRL Structures [6]

2.2.2 Policy Gradient-based DRL

Policy gradient-based DRL algorithm is another type of DRL that depend on optimizing policies regarding the expected return, such as expected cumulative reward, by gradient descent. This type of DRL utilizes gradient theorems to obtain optimal policy parameters. The estimation of a value function from the current policy is usually required by policy gradient, which can be achieved by utilizing actor-critic architecture. The policy structure is the actor, for it is to select actions, and the estimated value function is the critic, as it criticizes the actions conducted by the actor [45]. [46] demonstrated that even in a standard CPU-based computer environment, asynchronous execution of multiple parallel agents can learn efficiently in terms of time and resources. They proposed asynchronous advantage actor-critic (A3C) architecture, which is the asynchronous version of actor-critic. The results

showed significant performance in both 2D and 3D games with continuous domains and discrete action spaces.

2.2.3 Value-based DRL

[47] built the most well-known value-based DRL algorithm, the Deep Q-network (DQN), that can take and learn from high-dimensional inputs directly. DQN adopts the structure of CNNs to define a policy by estimating a value function $Q(s, a)$. DQN improves the stability of learning by using mainly four techniques: experience replay, target network, clipping rewards, and skipping frames [48]. [49] introduced Double DQN (DDQN) to make up for the possible upward bias caused by DQN with two estimators: one for selecting an action, and one for evaluating an action [6]. [50] indicated that DQN and DDQN perform significantly better if critical experience transitions are emphasized and replay them more frequently.

2.3 Audio Data Pre-Processing and Augmentation

2.3.1 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 [51][16]. 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 [52].

In many audio-based applications, such as automatic speech recognition (ASR) and acoustic event detection (AED), background noises sometimes overshadow the foreground sound events. [53] 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. [54] 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 Meidan filter is used to remove the drift of the mel spectrogram.

Commonly-used pre-processing methods for ASC applications are filtering methods. [8] 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 [55]. 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 [51].

2.3.2 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 [51], such as ImageNet [56]. 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 [57]. 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 [58]. [59] 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 [60] 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. [61] 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) [38]. 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 [51].

2.4 Recent Research on DL-based Audio Application

This section reviews the literature on audio-based DL applications. The literature is usually divided into different categories: sound classification, automatic speech recognition, spoken dialogue systems, emotions modeling, audio enhancement, music generation, and more [16]. In this section, we will only focus on automatic speech recognition, audio enhancement, music generation, and sound classification. Table **??** is a summary of reviewed research papers in audio-based DL applications.

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 [62][63][64]. [62] 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. The other DL model, sequence-to-sequence models, has demonstrated significant success in ASR. But the issue with the Seq2Seq model in ASR was the interference in real-world speech. [63] proposed a solution for this by using a sequence-to-sequence model trained with a policy

gradient algorithm. The proposed system indicated remarkable improvement in real-world scenarios with the combination of an RL-based and a maximum likelihood estimation (MLE) objectives, instead of training with the MLE objective alone. There has always been an issue of semi-supervised training with Seq2Seq ASR models. [64] proposed a REINFORCE algorithm for ASR. The algorithm rewarded the ASR to produce more correct sentences for the input data of both paired and unpaired speech. The proposed system achieved lower character error rates of 8.7%.

Audio-based intelligent systems are extremely sensitive to environmental noise, and the system's accuracy decreases when the noise level goes up [65]. 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 [66]. [66] 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. [67] 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 [68][69]. 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 [68]. [69] proposed an LSTM-based model that can compose polyphonic music based on music theory with better quality. [68] 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 [70], environmental sound classification [58], music classification [71], and more. [70] 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. [58] 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. [71] 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.

2.5 Evolution of Audio Features

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 [16]. The evolution of audio features can be divided into four categories: time domain features, frequency domain features, joint time-frequency domain, and deep features [7]. Figure 2.3 shows the evolution of audio 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 [72][73]. Time domain features were widely used

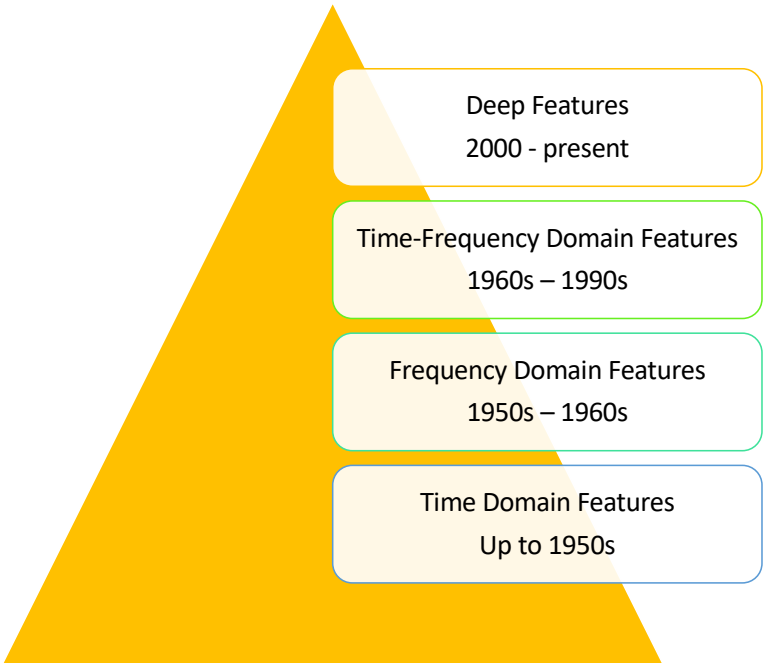


Figure 2.3. Evolution of Audio Features

in audio/acoustic analysis and audio-based classification since then [7]. Between the 1950s to 1960s, frequency domain features, like formant and pitch, were discovered and adopted in different audio-based applications since then [74][75]. 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 [76]. 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 [77][78], audio/video analysis[79], and speaker recognition [80], since 2010.

2.6 Feature Extraction Methods for Audio

2.6.1 Window Function

The most straightforward way to analyze an audio signal is via its original form [7] [16]. All the audio signals discussed are time series signals, which means signals that develop over

Table 2.1. Time Domain Features

Feature	Popular audio-based applications	Paper
Zero-crossing rate (ZCR)	music/speech discrimination, music genre classification, voice activity detection and vowel detection and analysis	[81][82][83][84]
Amplitude descriptor (AD)	environmental sound classification	[85]
Attach Delay Sustain Release (ADSR) envelop	music genre classification	[86]
Log attack time (LAT)	music genre classification	[87][88]
Shimmer	stress and emotion classification[89], speaker detection and verification, music sound classification	[89] [90][91]
Short time energy (STE)	environmental sound detection, audio-based surveillance systems, music onset detection, vowel detection and analysis	[92][93][94][84]
Volume	acoustic scene classification, speech and music classification, speech segmentation	[95][96]
Temporal centroid (TC)	environmental sound classification, acoustic scene classification	[97] [98].
Auto-correlation based features	acoustic scene classification, music tempo and beats estimation	[99][100]
Rhythm-based features	music genre classification, music instrument classification, speech/music discrimination, analysis of pathological speech	[101][96][102]

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 can not 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 [7]. 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 [103]. 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 [104]. 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 [103]. These window functions are at 0 on the edge of the window but increase gradually to become 1 in the middle of the window.

2.6.2 Time Domain Features

Table 2.1 is a summary of the selected time domain features, with their applications in audio and sound processing.

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 [105]. ZRC 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 [81]. Thus, ZCR can provide indirect information about the frequency of the audio signal. ZCR has been used to develop

classifier and discriminator [82], music genre classification, voice activity detection [83], and vowel detection and analysis [84].

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 [7]. 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) [85]. AD has mainly been used in environmental sound classification and animal sound classification.

Attach Delay Sustain Release (ADSR) Envelop

The Attach Delay Sustain Release (ADSR) envelop is another type of amplitude-based feature. The ADSR envelope feature doesn't work in real-time environmental sound classification systems for the delay part is not clearly showing and its sustain part is not showing in speech and environmental sounds. ADSR envelop features have been used in music analysis and music genre classification [86].

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[106]. It has been used in environmental sound classification [87] and music onset detection [88].

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 [107]. It has been used in stress and emotion classification [89], speaker detection and verification [90], and music sound classification [91].

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 [108]. The number of STE is high for the voiced segment and low for the unvoiced segment. STE has been used in environmental sound detection [92], audio-based surveillance systems [93], music onset detection [94], and vowel detection and analysis [84].

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 [95], speech and music classification [96], 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 [97] and acoustic scene classification [98].

Auto-correlation Based Features

The autocorrelation is the correlation of a signal with a delayed replica of itself, as a function of delay [7]. 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 [99], and music tempo and beats estimation [100].

Rhythm-based Features

The rhythm defines as a strong, regular, repeated pattern of a sound over time [109]. 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 [7]. Rhythm-based features have been used in

Table 2.2. Frequency Domain Features

Feature	Popular audio-based applications	Paper
Linear Predictive Coding (LPC) Coefficients	audio retrieval and audio segmentation	[110]
Code Excited Linear Prediction (CELP)	environmental sound classification	[111]
Linear spectral frequency	speaker segmentation, voiced/unvoiced detection, speech/music classification	[112][96]
Peak frequency	music and speech classification, gender classification	[101][113]
Spectrum envelope	music genre classification, environmental sound classification	[114][92]
Chroma based features	music genre classification	[115]
Spectral Centroid	music classification, music mood classification	[114][116]

music genre classification [101], music instrument classification, speech/music discrimination [96], and analysis of pathological speech [102].

2.6.3 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 [117]. Frequency domain features are the most important ones in audio signal analysis and processing [7]. Table 2.2 shows the selected frequency domain features and their applications.

Linear Predictive Coding (LPC) Coefficients

The linear Predictive Coding (LPC) Coefficients is one of the auto regression-based features, which are extracted from linear prediction analysis of a signal. LPC eliminates the redundancy from a signal and makes predictions on the next value by combining the

previous known coefficients [7]. LPC is an all-pole filter that uses a linear prediction model to represent the spectral envelope of digital speech in compressed form. It has been used in audio retrieval and audio segmentation [110].

Code Excited Linear Prediction (CELP)

The code excited linear prediction (CELP) is another type of auto regression-based feature. There are three techniques supported CELP. The first one is to use a linear prediction model to imitate the vocal tract. The second one is to use adaptive or fixed codebook entries as stimulation signals to the linear prediction model. The last one is to search in a closed-loop and perceptually weighted environment. CELP is a speech coding algorithm that provides better quality than lower bitrate algorithms such as linear predictive coding vocoders and residual dropout linear prediction algorithms. It has been used in environmental sound classification [111].

Linear Spectral Frequency

The linear spectral frequency is also a type of auto regression-based feature. It has another name called linear spectral pairs. LSF is used to indicate the linear prediction coefficients for the signal transmission on the channel. The linear prediction polynomial is expressed as the average of the palindrome polynomial and the inverse palindrome polynomial. The roots of the palindromic and anti-palindromic polynomials are conjugate in nature, so half of the roots are transmitted. The LSF characterization of a linear prediction polynomial contains only the locations of the roots. LSF indicates the variation in numbers when the glottis is open or closed. It can be used in speaker segmentation [112], voiced/unvoiced detection, and speech/music classification [96].

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 [7]. It has been used in music and speech classification [101] and gender classification [113].

Spectrum Envelope

The spectrum envelope is the logarithmic frequency power spectrum of a signal and can be used to generate a simplified spectrogram of an audio signal. The spectral envelope generated by the linear prediction method is called the linear prediction spectral envelope. The spectral peaks of the audio signal are more accurate due to the error optimized by linear prediction and the envelope is emphasized as in the auditory system. The spectrum envelope has been used in music genre classification [114] and environmental sound classification [92].

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 [115].

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 [114], music mood classification [116].

Table 2.3. Cepstral Domain Features

Feature	Popular audio-based applications	Paper
Mel Frequency Cepstral Coefficients (MFCCs)	surveillance-related events, environmental sound classification, speech recognition, speech enhancement, speaker recognition, music genre classification	[1][118][119][120][121][122]
Linear Prediction Cepstral Coefficients (LPCCs)	noise removal, music classification, speech recognition, speech analysis	[123][124][125][126]
Perceptual Linear Prediction (PLP) Cepstral Coefficients	speech recognition, environmental sound classification, emotion recognition	[127][128][129]
Relative-spectral PLP (RASTA-PLP) feature	peech recognition, speaker verification, gender classification	[130][131][132]
Greenwood function cepstral coefficients (GFCC)	environmental sound classification	[133]
Gammatone cepstral coefficients (GTCCs)	environmental sound classification, speech recognition	[134][135]

2.6.4 Cepstral Domain 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 2.3 shows the selected cepstral features and their applications.

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. From the above steps, a cepstrum created over Mel frequencies is called MFCCs [136].

MFCC is one of the most widely used features in audio processing applications like surveillance-related events [1], environmental sound classification [118], speech recognition [119], speech enhancement [120], speaker recognition [121], music genre classification [122], and more.

Linear Prediction Cepstral Coefficients (LPCCs)

Cepstrals have many great properties, such as source filter separation, orthogonality, compactness, and more. These make cepstral coefficients reliable and suitable for machine learning. In addition, the Linear Prediction Coefficient (LPC) is very sensitive to numerical precision, so it's necessary to transform the LPC to the cepstral domain. The transformed coefficients are called Linear Prediction Cepstral Coefficients (LPCCs). LPCC has been used

in noise removal [123], music classification [124], speech recognition [125], speech analysis [126], and more.

Perceptual Linear Prediction (PLP) Cepstral Coefficients

The Perceptual Linear Prediction (PLP) coefficients originated from Linear Prediction Coefficients (LPC). PLP Coefficients are proceeded by perceptual processing before autoregressive modeling. From the above process, the linear coefficients are converted to cepstral coefficients. PLP cepstral coefficients have been used in speech recognition [127], environmental sound classification [128], and emotion recognition [129].

Relative-spectral PLP (RASTA-PLP) feature

The RASTA-PLP is an extended version of perceptual linear prediction (PLP). RASTA-PLP is obtained by applying a bandpass filter to each subband. The added filter helps to smooth out short-term noise variations and remove any constant shifts in the speech signal caused by static spectral coloration [137]. This feature has been used in speech recognition [130], speaker verification [131], and gender classification [132].

Greenwood function cepstral coefficients (GFCC)

GFCCs were introduced as a generalized form of MFCCs. GFCC uses mel-scale features and it is theoretically applicable to almost all land mammals and provides good vocal performance for almost all species. GFCC can be implemented using very basic knowledge of the minimum and maximum frequency ranges for a particular species and is derived from Greenwood's equations. The Greenwood equation maps the cochlear frequency locations in almost all species. This feature has been used in environmental sound recognition, especially for animal and bird sound classification [133].

Gammatone cepstral coefficients (GTCCs)

Noise reduction or cancellation is still the main challenge for automatic speech recognition (ASR) systems. Recent studies indicated that Gammatone cepstral coefficients (GTCCs) shows robust performance against noise in many ASR systems and applications. GTCC is based on gamma-pass filter banks that take as output a cochlear map, which is actually a frequency-time representation of the sound signal. The extraction process of GTCCs is similar to MFCCs, except that the gamma tone filter bank is used instead of the mel-filter bank. GTCCs have been used in environmental sound classification [134] and speech recognition [135].

2.7 Feature Extraction Tools

There are many tools available for audio feature extraction that comes in different formats. There are mainly three different kinds: software function libraries, plug-ins for a host application, and stand-alone software applications.

Librosa is a Python library for music and audio processing. There are four different categories under feature extraction when using Librosa, which are spectral features, rhythmic features, feature manipulation, and feature inversion [138].

Marsyas (Music Analysis, Retrieval and Synthesis for Audio Signals) is an open-source software framework for music and audio processing with an emphasis on music information retrieval. It is a C++ library. One of the most powerful executables provided by Marsyas is bextract, which can be used in real-time music classification [139].

jAudio is a software package for audio feature extraction that is designed to make the process of feature calculation and extraction easier and faster. jAudio is a java-based software and it supports various types of outputs, such as XML format and the ARFF format [140].

Aubio is a software designed for high-level feature extraction in audio and music processing. The functions include file segmentation, pitch detection, beat tapping, and more [141].

Essentia is an open-source C++ library for audio analysis and audio-based music information retrieval. It provides a large number of reusable algorithms that implement audio input/output functions, standard digital signal processing modules, statistical representations

of data, and a large number of spectral, temporal, tonal, and high-level musical descriptors [142].

Libxtract is an open-source software library for feature extraction in audio processing. The purpose of creating this library is to provide a superset of the MPEG-7 and Cuidado audio features. Libxtract is written in C, but can be used in many other platforms and computer languages [143].

YAAFE is an audio feature extraction software written in C++. YAAFE is known for its efficiency in feature calculation with low computation cost. It is easy to configure and each feature can be set independently [144].

Meyda Web Audio API based low-level feature extraction tool, written in Javascript. Designed for web browser-based efficient real-time processing [27].

MIRToolbox is a Matlab toolbox that extracts musical features from audio files. It can be used for statistical analysis, segmentation, and clustering [145].

The **Timbre Toolbox** is another toolbox in Matlab for audio and music processing. It extracts different kinds of features that can be used in machine learning methods for music information retrieval, perception research, and content-based retrieval using large sound databases [146].

2.8 ML-Based UAV Classification with Computer Vision

In recent years, machine learning-based computer vision has been a popular approach used in many research fields and applications, such as autonomous driving, medical diagnosis, facial recognition, and object detection and classification. Vision-based machine learning has also been adopted in UAV detection and classification with promising results [4]. Depending on which kind of feature extraction methods the researchers choose to use, vision-based UAV detection and classification can be divided into two categories: the first one is to feed the machine learning models with learned features that are extracted from image or video data, and the second one is to use hand-crafted low-level features. Research papers on ML-based UAV detection using computer vision are summarized in Table 2.4.

Table 2.4. Research Papers on ML-based UAV detection with Computer Vision

Research	Database	Classification objects	Feature types	ML models	Results
[147]	Artificial dataset	UAVs	Learned features	CNN, YOLO	0.9 of precision and recall scores
[148]	UAV and Aircraft database	UAVs and Aircraft	Learned features	CNN, boosted trees	0.849 and 0.864 of precision score for UAV and Aircraft databse, respectively
[149]	10000 Google images	UAVs	Learned features	CNN	89% and 91.6% in detection and identification, respectively
[150]	Bird-Vs-Drone dataset	UAVs	Learned features	CNN, VGG and ZF	0.66 of mean average precision
[151]	UAV and video-based bird database	UAVs and Birds	Learned features	Recurrent Correlational Networks	0.540 of missing rate
[152]	Synthetic database using Physically Based Rendering Toolkit (PBRT)	UAVs	Learned features	Faster R-CNN, ResNet-101	80.69% of mean average precision
[153]	Self-collected UAV and bird dataset	UAV vs Birds	Hand-crafted features, Generic Fourier Descriptor	Neural Network	85.3% of accuracy

[147] proposed an end-to-end object detection model based on CNN to detect UAVs. They used YOLO, which is an extension of the existing CNN model and the state-of-the-art real-time, single-shot object detection network. They combined real drone and bird images with different background videos to create a large artificial dataset to train and evaluate their model. They achieved high precision and recall scores of 0.9 at the same time.

[148] proposed a regression-based approach for object-centric motion stabilization of image patches that outperforms state-of-the-art techniques. Their method provided an effective classification of spatiotemporal image cubes. They applied two different methods to detect UAVs from a single moving camera: one based on boosted trees and the other one based on CNN. The authors collected their own database with two categories: UAV and Aircraft datasets, to train and evaluate their models. They achieved average precision of 0.849 and 0.864 for UAV and Aircraft categories, respectively.

[149] proposed a UAV detection system using UAVs with cameras. Their dataset includes about 7000 UAV images from Google and about 3000 non-UAV images. Their system consists of two parts: a drone detection module and a drone identifier module. The drone detection module used HARR feature-based classifier from the OpenCV library, and the drone identifier module used a simple CNN with two convolutional layers and two fully connected layers. They achieved 89% accuracy in detection and 91.6% in identification.

[150] compared different CNN-based network architectures, such as Zeiler and Fergus (ZF) and Visual Geometry Group (VGG16) to detect UAVs from video data. They used transfer learning to train their networks with pre-trained models for insufficient training data. They used the Bird-Vs-Drone dataset to train and evaluate their models. 5 MPEG4-coded videos are included in the dataset, with total 2727 frames. Their results show that VGG16 with Faster R-CNN has the best performance and achieved 0.66 of average precision.

[151] proposed a single, trainable, end-to-end neural network called Recurrent Correlational Network that detects and tracks small birds and UAVs. They used a video-based bird dataset and the UAV database collected by [148]. The proposed network is divided into four modules: convolutional layers, ConvLSTM layers, a cross-correlation layer, and a fully-connected layer for object scoring. It is worth mentioning that instead of training from scratch, they adopted

Table 2.5. Research Papers on ML-based UAV detection with Radar

Research	Radar System	Data	Classification objects	Feature types	ML models	Results
[154]	S-band CW Doppler radar	280 images	UAVs	Spectral correlation function (SCF)	Deep Belief Network (DBN)	above 90% in accuracy
[155]	Ku-band FMCW radar	50000/10000 indoor/outdoor images	UAVs	Contatendated MDS and CVD	CNN	94.7% in accuracy
[156]	K-band and X-band CW radar	720 samples per radar	UAVs	PCA-based features	SVM	94.7%
[157]	Ka-band CW radar	30*10 seconds trials per UAV, simulated bird MDS	UAVs vs Birds	Mean spectrogram, SVD, CVD	SVM	96% to 100% in accuracy
[158]	S-band BirdRad	800 trail samples	UAVs vs Birds	9 polarimetric features	Nearest-neighbour classifier	100% in accuracy

a fine-tuning approach by using AlexNet and VGG16. They reported the results in ROC curves, and their approach outperformed the existing methods.

[152] proposed to use of the Physically Based Rendering Toolkit (PBRT) to render a large number of synthetic UAV images with variations. In a way, the UAV images in their dataset are indistinguishable from the real ones. The rendered image provides the bounding boxes, the locations of important parts, and the locations of all pixels of UAVs. They trained the Faster R-CNN for UAV detection with the rendered dataset and evaluated it on a test dataset that contains manually annotated real UAV images. The Faster R-CNN they trained achieved 80.69% of average precision, which is much higher than the one trained PASCAL VOC 2012 dataset.

[153] proposed a vision-based feature, Generic Fourier Descriptor (GFD), with a neural network to classify UAVs versus birds. They created their own dataset with 930 bird and 410 UAV images from open source and used 5-fold cross-validation. They created a neural network with about 10000 neurons to classify GFD features for birds and UAVs. They achieved an 85.3% classification rate with the whole dataset and 93.1% with the subset of 162 images from the whole dataset.

Vision-based UAV detection and classification usually require a camera that is relatively low cost, and computer vision with machine learning has been a popular and fast-growing research area. Current research shows promising results. Most of them rely on learned features along with different machine learning models [147]–[152]. But the problem remains that there is few publicly available dataset for training and evaluation. The other challenge for vision-based UAV detection is the noise in visual data caused by heavy rain, snow, or other types of natural factors.

2.9 ML-Based UAV Classification with Radar

Radars are devices that radiate electromagnetic energy and detect the echo returned from reflecting objects (target)[159]. Based on the echo returned, radars collect information about the position and nature of the target. The capacity of these devices to detect an object is highly conditioned by the targets Radar Cross-Section (RCS), an attribute of objects that

describes the intensity of the echo they return when exposed to an electromagnetic wave, and which depends on the physical attributes of the object, such as composition, size, shape, radiation, and polarization, among others [159].

So far, the most popular approach for UAV detection is radar recognition [4]. Current and previous research applied different radar front ends, feature extraction methods, and ML models in radar UAV detection, and their detection and classification results are promising. Research papers on ML-based UAV detection using radar are summarized in Table 2.5.

[154] proposed a system using S-band continuous wave (CW) radar with a deep belief network (DBN). From the micro-Doppler signature (MDS), the system first extracts the spectral correlation function (SCF), which is used to train the DBN to classify UAVs. Three micro-drones are used in their research, which are an artificial bird, a helicopter, and a quad-copter. There are a total of four classification classes with three UAV classes and a reference class. The DBN extracted 70 SCF images from each class. As for the data augmentation, they added different levels of Gaussian noise to 50 out of 70 images for each class. They achieved above 90% in accuracy.

[155] proposed a UAV classification system by using CNN and MDS. They adopted a pre-trained CNN, GoogleNet, for training and testing. They used a frequency-modulated continuous-wave (FMCW) radar and two UAVs, an Inspire I and an F820, in their experiments. The dataset consists of images that merged from MDS and CVD, which are generated by Ku-band FMCW signal. 53410 and 13560 images are generated from an anechoic chamber (indoor) and outdoor environment, respectively. They tested their system with different motors and different angles of a UAV. Their system achieved 94.7% in accuracy. Classification of two different UAVs at two different heights f 50 and 100 meters results in 100% accuracy.

[156] proposed a dual-band radar classification system with a K-band and an X-band CW radar. Three different UAVs are used in the experiment: a quadcopter, a helicopter, and a hexacopter. The two radar sensors collected the time-frequency spectrogram by conducting a short-time Fourier Transform (STFT) on the radar data. Then PCA-based (principal components analysis) features are extracted from the spectrogram by the two radar sensors. The features from the two sensors are then merged together. The extracted and merged features are used to train the SVM. Their system has collected 720 samples from each radar

Table 2.6. Research Papers on ML-based UAV detection with Radio Frequency

Research	Classification objects	Feature types	ML models	Results
[160]	UAVs	Hash fingerprint	SDVV	Positive detection results in indoor environment
[161]	Parrot Bebop, DJI Phantom	UVAs Reflected signal from propellers, the signal between the controller and the UAVs, and body vibration	Wavelet analysis and maximum PSD	Positive detection results
[162]	UAVs vs non-UAVs	Skewness variance, entropy and kurtosis with NCA	SVM, DA, ANN, and KNN	96.3% with KNN, 96.84% with SVM in classification accuracy
[163]	UAVs	Self-adaptive threshold	GMM	Above 97% in classification accuracy

sensor for each UAV. Their results proved that the dual-band classification system performs better than the radar system with only one sensor. The dual-band system achieved up to 94.7% in classification accuracy.

[157] proposed a Ka-band CW radar system to classify UAVs versus birds and different UAVs. Their system used different UAVs to collect data: four quadcopters of different sizes, an octocopter, a small helicopter and a fixed-wing plane (Multiplex Funcub). The radar system extracts three different kinds of features, which are a time-averaged spectrogram, the first left singular vector of singular value decomposition (SVD), and a mean cadence velocity diagram (CVD). They generated the bird flying data by using the same CW radar configuration to collect UAV data. They used SVM to train and test the proposed system. They achieved 96% to 100% in classification accuracy.

[158] proposed to use polarimetric features to classify UAVs versus large birds. Sometimes, the radar system confuses those two because of their similar RCS and motion patterns.

They collected 8000 data points from two UAVs by using BirdRAD, which is an S-band radar system at 3.25 GHz. The nine features extracted include linear depolarization ratio, differential depolarization ratio, co-polarized correlation coefficient, cross-polarized correlation coefficient, entropy, anisotropy, polarimetric eigenvector, and orientation angle. The features are used to train and test on a nearest-neighbor classifier. The system achieved nearly 100% in classification accuracy for close-range tests (300-400 meters).

With different types of radars, different features extracted, and ML models, the ML-based radar approach for UAV detection and classification shows promising results. However, the problem remains whether the systems can be applied to different types of UAVs, different types of radar sensors, or to cover longer distances of detection environment. Also, radar system has difficulty detecting a certain type of UAVs, such as micro-UAVs, because of their small Radar Cross-Section (RCS) and low altitudes flying patterns [164].

2.10 ML-Based UAV Classification with Radio-Frequency

Radio-frequency is another approach used in ML-based UAV detection and classification. UAVs and their controllers communicate and exchange signals via control signals at the well-known 2.4GHz radio band (ISM band). Radio frequency signals can be used to detect not only the UAVs but also the controller of the UAVs [4]. Research papers on ML-based UAV detection and classification using radio-frequency are summarized in Table 2.6.

[160] proposed a distance-based support vector data description (SVDD) system to classify low, slow, and small UAVs that use 2.4GHz frequency. The system generates features called hash fingerprints to train the SVDD. They have collected the data themselves to train and evaluate their system. The results show that the system can successfully detect UAVs in the indoor environment. However, the detection false rate increases in the outdoor environment, because they added the white Gaussian noise (WGN) to the original signal.

[161] proposed a UAV detection system that includes active and passive modes. The active mode detects UAVs by constantly listening to the reflected wireless signal, and the passive mode is to observe the communication between the controller and the UAVs. Instead of transmitted signals, the proposed system is based on physical features: the reflected signal

from UAV's rotating propellers, the communication between the controller and the UAVs, and the vibration from UAV's main body. They used two UAVs, Parrot Bebop and DJI Phantom, to evaluate the system. Their results show that the proposed system can detect UAVs by listening to the wireless signal in the area that it passes through.

[162] proposed to use radio frequency fingerprints of the signal transmitted between the controller and the UAV to detect and classify UAVs. The detection system adopted a Bayesian approach based on the Markov models to classify UAVs versus non-UAVs. And the classification system used the radio frequency signal to classify 14 different UAVs. The features extracted from the RF signals are skewness, variance, energy spectral entropy, and kurtosis. They used neighborhood component analysis (NCA) as the feature selection method. The selected features then were fed into several ML models, including KNN, discriminant analysis (DA), SVM, and neural networks (NN). They evaluated the proposed method on a dataset of 100 RF signals collected by 14 different UAV controllers. They achieved 96.3% and 96.84% in classification accuracy with KNN and SVM, respectively.

[163] proposed a UAV detection system with an adaptive threshold based on Gaussian mixture model(GMM). The system can detect the start point of the signal source. They used 7 different UAVs to collect data, which are DJI 3 pro, DJI 4, DJI Mavic pro, Mi, Hubson, Xiro, and Phantom 4 pro. They collected 2000 data samples per UAV. GMMs of the collected signal are calculated by using Expectation Maximization Algorithm (EM). Then the threshold τ is obtained by selecting the proper Gaussian destitution. If the sampling points are greater than the threshold τ, the start-point of the UAV signal is detected. Their results show 97% detection accuracy.

The radio-frequency signal is an important feature of UAVs that can be used to in UAV detection, classification, and localization. But RF-based UAV detection system has a very low or zero accuracies rate when the UAV is in autonomous flying mode [4]. When UAVs fly autonomously, they usually follow the pre-calculated GPS points. Thus, there will be limited RF-based signal communication between the controller and UAV. Also, applying machine learning techniques with RF-based data is not very common, and there hasn't been much done in this area of research. RF-based research also needs a more publicly available dataset.

Table 2.7. Research Papers on ML-based UAV detection with Audio

Research	Classification objects	Feature types	ML models	Results
[165]	UAV vs nature daytime vs street with traffic vs train passing vs crowd	Short-time energy, temporal centroid, Zero Crossing Rate (ZCR), spectral centroid, spectral roll-off, and Mel Frequency Cepstral Coefficients (MFCCs)	SVM	96.4% in classification accuracy
[166]	UAVs vs non-UAVs	Normalized STFT	CNN	98.97% in classification accuracy
[167]	UAVs vs nature background vs rain	Power spectrum density (PSD)	SVM	Best performance with signal-to-interference ratio (SIR) greater than 10 dB
[168]	UAVs	MFCC, STFT	SVM, CNN	promising detection results reported in color map
[1]	UAVs	MFCCs, chroma, mel, contrast, tonnetz	SVM, Gaussian Naive Bayes (GNB), KNN, Neural Network (NN)	Above 95% with combination features
[169]	UAVs	MFCCs	Gaussian Mixture Model (GMM), CNN, RNN	80% in classification accuracy with RNN
[170]	UAVs	Spectrum images, Fast Fourier Transform (FFT)	Correlation, KNN	80% in classification accuracy with image correlation

The other challenge is whether others can reproduce from the existing work with different models of UAVs.

2.11 ML-Based UAV Classification with Audio

When a UAV is flying, it emits a continuous humming sound. Ideally, different types and models of UAV project different sounds that can be identified as audio fingerprint [4]. Many researchers have used the sound of the UAV in detection and classification. Research papers on ML-based UAV detection using audio data are summarized in Table 2.7.

[165] proposed a multi-class SVM structure to detect UAV and other signals using audio data. The other classes include the sound of nature daytime, street with traffic, train passing, and crowd. They collected their own dataset focusing on the audio input that is higher than 48kHz. The dataset consists of seventy minutes of audio for each class. Six features were extracted from the collected audio data, including short-time energy, temporal centroid, Zero Crossing Rate (ZCR), spectral centroid, spectral roll-off, Mel Frequency Cepstral Coefficients (MFCCs). The extracted features were used to train SVM. They achieved 96.4% in classification accuracy in detecting UAVs.

[166] proposed to detect UAVs by using the normalized STFT from UAVs audio data. The normalized STFT was extracted as the feature of the audio data. They also collected their own dataset by using two UAVs, a DJI Phantom 3 and a Phantom 4. The feature extracted from the collected dataset were used to train a CNN and data for evaluation used the collected dataset with Additive white Gaussian noise (AWGN). The dataset consisted of 68931 UAV audio frames and 41958 non-UAV frames. They achieved 98.97% in classification accuracy, and they reported a false alarm rate of 1.28.

[167] proposed an acoustic wireless sensor network (WSN) to detect and localize UAVs. Authors discovered that a unique feature of UAV from other natural sounds is power spectrum density. They collected their own dataset for three different classes which are UAVs, natural background, and rain. Each class consisted of 2000 audio samples. They added additional Gaussian noise to the data during testing. They reported that their system can successfully detect UAVs and the best performance with a signal-to-interference ratio (SIR) greater than 10 decibels.

[168] proposed a multi-node acoustic system to detect UAVs. The UAV used to collect data was Parrot AR Drone 2.0. Two features extracted from the audio data are MFCCs

and STFT. The authors collected data in two categories: UAVs flying from 0 to 10 meters above the acoustic system; environmental noise from the same location where the UAV data was collected. They trained two ML models, SVM and CNN, with extracted features. They reported positive classification results in the form of color maps.

[1] evaluated different feature extraction methods for audio data and proposed a feature-based UAV classification system. The authors evaluated five feature extraction methods, available in the Python Librosa library, including MFCCs, chroma, Mel-spectrogram(mel), contrast, and tonnetz. A DJI Phantom 4 and an EVO 2 Pro were used to collect audio recordings. They collected 300 samples for each type of UAV, and 600 samples for the environmental noise, with each sample about 10 seconds. Five individual features and one combination of five were used to train four machine learning models, which are NN, SVM, Gaussian Naive Bayes (GNB), and KNN. The results showed that the use of a combination of features improves the accuracy greatly, compared to individual features. They achieved above 95% in classification accuracy with combination features.

[169] proposed to use GMM, CNN, and RNN models to detect UAVs in real-time that are within 150 meters. They used the augmentation technique to enlarge the dataset by adding different environmental sounds to the raw UAV audio data. Different UAVs were used in training and testing. The two features extracted from the audio data were MFCCs and mel-spectrogram. The results showed that the audio data collected from more than 150 meters away did not provide much useful features/information. They achieved 80% in classification accuracy with RNN, and the processing time was 240 milliseconds.

[170] proposed a real-time UAV detection and monitoring system. The system used Fast Fourier Transform (FFT) and generated spectrum images of the audio data. They used two UAVs, Phantom 1 and 2, to collect UAV audio data. The dataset included 70 audio samples from each UAV. They used the FFT to train two ML models, which were Plotted Image Machine Learning (PIL) and KNN. PIL used the spectrum image data from FFT, and KNN used csv files with FFT-format data. They achieved 83% classification accuracy with PIL and 61% with KNN.

There have been an increasing number of research in ML-based UAV detection using audio data. Current and previous research showed promising results by using an audio

Table 2.8. Research Papers on ML-based UAV payload detection

Research	Payload Detection System	Data	Classification Objects	Feature Types	ML Models	Results
[171]	Audio classification	2544 samples	UAV with three different payloads	MFCC, Mel, Chroma, Tonnetz, and Contrast	CNN, RNN, CRNN	94.9% in with CNN and MFCC
[5]	Audio classification	1200 samples	loaded vs unloaded UAVs	chroma, mel, MFCCs, contrast, and tonnetz	SVM, GNB, KNN, NN	98% in average accuracy with combination features
[172]	S-band pulsed radar (NetRAD)	45 samples per category	UAV with three different payloads	Centroid and bandwidth of MDS	Naive Bayes and Discriminant analysis	90% to 100% in accuracy
[173]	NetRAD radar system	45 samples for each payload class	UAVs with five different payloads	MDS, spectral kurtosis, PCA	KNN	92.61% in average classification accuracy

approach. However, the publicly available dataset is missing, which caused every individual researcher to collect their own dataset with different types of UAVs. The proposed system produced promising results and may not work on a different dataset. The other challenge for an audio-based detection system is environmental noise.

2.12 ML-based UAV Payload Detection

There is only a limited number of research on the topic of payload detection for UAVs. Radar [172], [173] and audio [171][5] are the main approaches to classifying loaded and

unloaded drones. Research papers on ML-based UAV payload detection are summarized in Table 2.8.

[171] trained and evaluated three deep learning models to classify three different payloads on two different drones using audio data. In this research, Ku et al. collected audio data from two different UAVs, DJI Phantom 4 and Autel Evo2. They also used data augmentation techniques on the collected dataset. The feature extraction method used in their research includes MFCC, Mel, Chroma, Tonnetz, and Contrast. The trained systems can classify three different payloads, which are no payload, 1 payload, and 2 payloads. The deep learning models used in this research include CNN, RNN, AND CRNN. All three models achieved remarkable performance with MFCC, which the combination of MFCC and CNN has the highedt accuracy of 94.9%.

[5] explored five different feature extraction methods and a combination to classify whether a drone carries a payload. The five selected feature extraction methods were chroma, mel, mfcc, contrast, and tonnetz, and the combinations of all five were also applied and evaluated. Features of each audio recording under each category (loaded and unloaded) are calculated and saved. The four machine learning models used for training are SVM, GNB, KNN, and a Neural Network. Those saved features are used as input to train ML models. The dataset was collected and labeled by using two different brands and models of drones, DJI Phantom 4 and EVO 2 Pro. The dataset consisted of 1232 audio samples of loaded and unloaded drones. The results showed that the combination of features has a better performance than individual ones. The combination feature achieved about 99% average accuracy in all four ML models.

[172] proposed a multi-static radar system, NetRAD, to classify different payloads of UAVs, including 0 gram (g), 200g, and 500g. They used quadcopter DJI Phantom Vision 2+ in collecting data. NetRAD extracted two features of centroid and the bandwidth of the MDS in 2-second windows. Three NetRAD nodes were used to collect data, and each node collected 15 samples per payload. The two classifiers they used are the Naïve Bayes and the diagonal-linear variant of the discriminant analysis. The proposed radar system achieved 90% to 100% in classification accuracy.

[173] proposed payload detection and classification system using NetRAD. The system used a new micro-Doppler feature extraction method to extract features including MDS,

and spectral kurtosis to classify five different payloads on UAVs. The extracted features were processed by PCA for dimensional reduction and then were fed into KNN model for training. They collected data with flying and hovering UAVs. Their system achieved 92.6% classification accuracy.

2.13 Summary

In this chapter, important architectures of Deep Learning are discussed, including CNNs, RNNs, Seq2Seq, and more. In addition, this chapter covers some popular DRL algorithms related to audio classification, including model-based DRL, policy gradient-based DRL, and value-based DRL. To overcome the lack of publicly available audio datasets, different approaches to data augmentation techniques are discussed in Section 2.3. Reviewed literature in Section 2.4 shows the advantages of using DL in audio and speech processing applications. A thorough review is presented in ML-based UAV detection and classification using different approaches, including radar, computer vision, radio-frequency, and audio. Papers on UAV payload detection are also discussed. Each approach has its advantages and disadvantages, but all with promising results. Furthermore, this chapter discusses the audio features that are usually used to train ML models from the time domain, frequency domain, time-frequency domain, and cepstral domain, including the evolution, characteristics, and applications of different features. Varies feature extraction tools in different formats are also discussed, which include software function library, plug-ins for a host application, and stand-alone software applications. Most of the tools are open-source and can be used on many modern architectures and platforms.

3. METHODOLOGY

3.1 Data Collection

In the data collection process, I fly and collect audio data from 22 different UAVs, as shown in Table 3.1. For each UAV, at least 100 audio entries are recorded and each audio entry includes 5 seconds of recording of the flying drone audio data. The manufacturers of the UAVs include most of the DJI, Autel, Syma, Yuneec, UDI, Hasakee, and self-build ones. Among the 22 UAVs, 20 of them are quadcopters. I operate and record UAVs by DJI, Autel Robotics, Yuneec, and self-built ones in the outdoor environment. All the UAVs by Syma, UDI, Hasakee, and 1 of DJI ones in the indoor environment. Figure 3.1 shows the estimated area where most of the outdoor data collection takes place. Most of the outdoor operation of drones took place in an open area in New Richmond, Indiana, and the coordinate is (40.2227062, -87.0000169). The indoor recordings took place in K-SW at Purdue University.

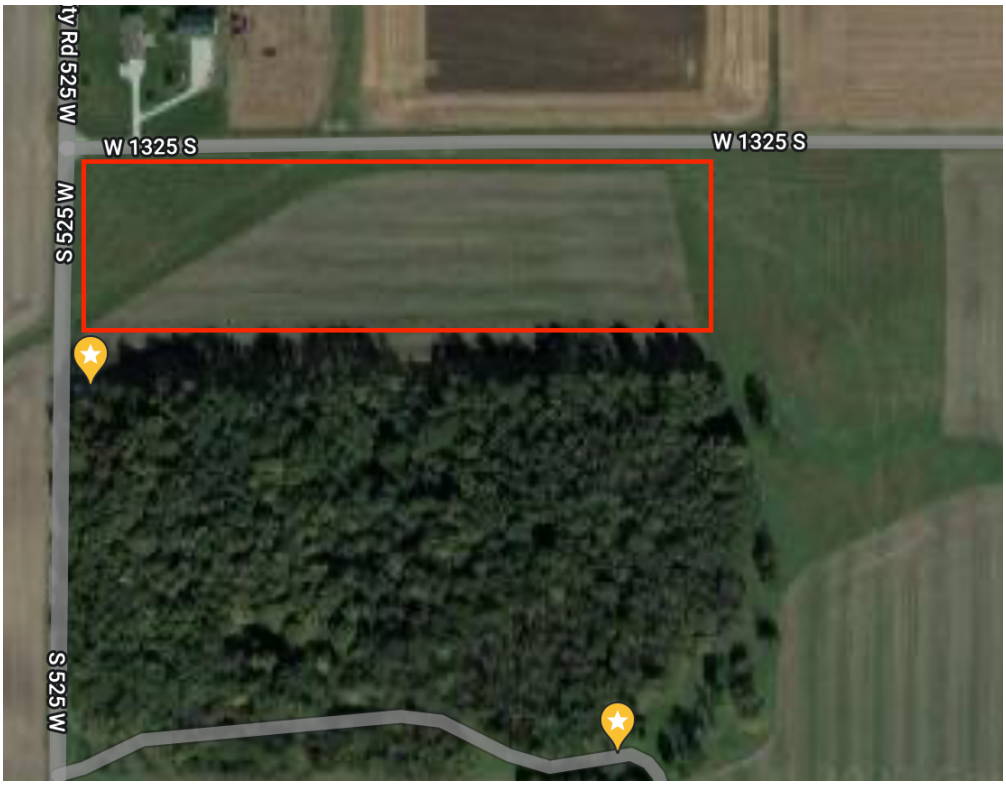


Figure 3.1. Outdoor Data Collection Site

This research also includes 1 tricopter and 1 hexacopter. There are two self-build UAVs included in the dataset, which are David Tricopter and PhenoBee[174]. David Tricopter is designed and built by my friend David Windestal. The flight control used for David Tricopter is AfroFlight Naze32. The weight of the tricopter including the battery is 2.6 lbs, and the diameter is 34 inches. As for the PhenoBee, it is built by my colleague, Ziling Chen, and it is the largest UAV in the dataset. PhenoBee weighs about 23kg, and the height and the diameter are 1.35 meters. The framework used for PhenoBee is Ardupilot, and the hardware is the Cubepilot Cube Orange.

Table 3.1. UAVs Included in the Dataset

Index	Manufacture	Model	Drone Type	Data Type
0	Self-build	David Tricopter	Tricopter	Outdoor
1	Self-build	PhenoBee	Quadcopter	Outdoor
2	Autel	Evo 2 Pro	Quadcopter	Outdoor
3	Yuneec	Typhoon H Plus	Hexacopter	Outdoor
4	Swellpro	Splash 3 Plus	Quadcopter	Outdoor
5	DJI	Matrice 200	Quadcopter	Outdoor
6	DJI	Matrice 200 V2	Quadcopter	Outdoor
7	DJI	Mavic Air 2	Quadcopter	Outdoor
8	DJI	Mavic Mini 1	Quadcopter	Outdoor
9	DJI	Mini 2	Quadcopter	Outdoor
10	DJI	Mavic 2 Pro	Quadcopter	Outdoor
11	DJI	Air 2s	Quadcopter	Outdoor
12	DJI	Phantom 2	Quadcopter	Outdoor
13	DJI	Phantom 4	Quadcopter	Outdoor
14	DJI	RoboMaster TT Tello	Quadcopter	Indoor
15	Hasakee	Q11	Quadcopter	Indoor
16	Syma	X5SW	Quadcopter	Indoor
17	Syma	X5UW	Quadcopter	Indoor
18	Syma	X20	Quadcopter	Indoor
19	Syma	X20P	Quadcopter	Indoor
20	Syma	X26	Quadcopter	Indoor
21	UDI RC	U46	Quadcopter	Indoor

3.2 UAVs Specifications

There are two self-built drones that are included in the dataset. David tricopter is built based on the design of David Windestal, which is shown in Figure 3.2. It uses a 3-propeller

design. The weight of the tricopter including the battery is 1.180 kg, and the diameter is 863.6 mm. The tricopter controls roll by rotating the rear propeller with a servo. The flight control used for David Tricopter is AfroFlight Naze32. h from David Tricopter is conducted in an outdoor environment in Columbus, IN.

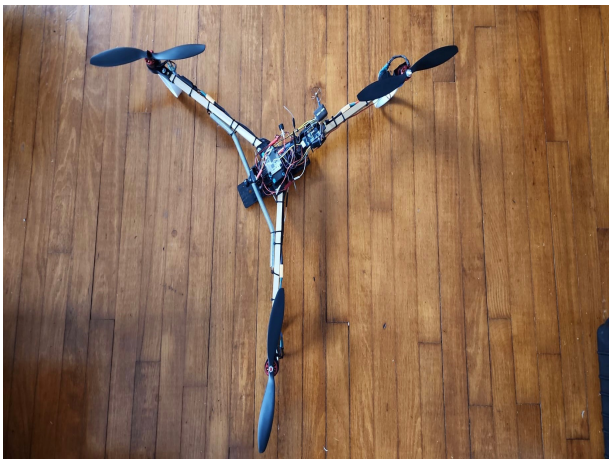


Figure 3.2. Self-build David Tricopter

Figure 3.3 shows the other self-built drone, PhenoBee, built by my colleague, Ziling Chen. It is so far the largest UAV in the dataset. PhenoBee weighs 23.000 kg, and the height and the diameter are 1350.0 mm. The framework used for PhenoBee is Ardupilot, and the hardware is the Cubepilot Cube Orange. The experiment of collecting audio data from PhenoBee and collected is conducted in Agronomy Center for Research and Education in West Lafayette, IN.

Autel Evo 2 [175] is shown in Figure 3.4. Evo 2 weighs about 1.130 kg, and the maximum take-off weight is 2.000 kg. The diagonal length of the wheelbase is about 396.0 mm. The experiment of collecting audio data from Evo 2 is conducted in the outdoor environment (Figure 3.1) in New Richmond, IN.

Figure 3.4 shows the Yuneec Typhoon H Plus [176], which is the only hexacopter in the dataset. Typhoon H Plus weighs about 1.645 kg without the camera and 1.995 kg with the camera. The diagonal length without propellers is about 520.0 mm. The experiment of collecting audio data from Typhoon H Plus is conducted in the outdoor environment (Figure 3.1) in New Richmond, IN

Figure 3.3. Self-build PhenoBee

Figure 3.4. Autel Evo 2

Swellpro Splash 3 Plus [177] is the only waterproof UAV in the dataset. It can land and fly on the water, with a waterproof level up to 23.6 inches deep for a short amount of time. Splash 3 Plus weighs about 1.447 kg without a battery, and the diagonal length is about 450.0 mm. The experiment of collecting audio data from Splash 3 Plus is conducted in the outdoor environment (Figure 3.1) in New Richmond, IN

DJI Matrice 200 [178] is a professional quadcopter, which is shown in Figure 3.7. Matrice 200 weighs about 3.800 kg with two standard batteries. The diagonal length of the wheelbase is about 643.0 mm. The experiment of collecting audio data from Matrice 200 is conducted

Figure 3.5. Yuneec Typhoon H Plus

Figure 3.6. Swellpro Splash 3 Plus

in the outdoor environment (Figure 3.1) in New Richmond, IN. Due to the expected wind condition, Matrice 200 crashed into tree bushes when landing.

Figure 3.8 shows the DJI Matrice 200 V2 [179]. Matrice 200 V2 is the improved version of 200. Matrice 200 V2 weighs about 4.69 with two standard batteries, which are heavier than 200. The diagonal length of the wheelbase is about 643.0 mm, which is about the same as 200. The experiment of collecting audio data from Matrice 200 V2 is conducted in the outdoor environment (Figure 3.1) in New Richmond, IN.

Figure 3.7. DJI Matrice 200

Figure 3.8. DJI Matrice 200 V2

DJI Mavic Air 2 [180] is shown in Figure 3.9. Air 2 weighs about o.570 kg, and the diagonal length of the wheelbase is about 302.0 mm. The experiment of collecting audio data from Air 2 is conducted in the outdoor environment (Figure 3.1) in New Richmond, IN.

Figure 3.10 shows the DJI Mavic Mini [181]. Mavic Mini weighs about 0.249 kg, and the diagonal length of the wheelbase is about 213.0 mm. The experiment of collecting audio data from Mavic Mini is conducted in the outdoor environment (Figure 3.1) in New Richmond, IN.

Figure 3.11 shows the DJI Mini 2 [182]. Mini 2 weighs a little less than 0.242 kg, and the diagonal length of the wheelbase is about 213.0 mm. The appearance of the Mini 2 and

Figure 3.9. DJI Mavic Air 2

Figure 3.10. DJI Mavic Mini

Mavic Mini are very similar. When detecting and classifying these two types, audio data would be the ideal approach, for the appearance are too similar for a classifier trained with image data. The experiment of collecting audio data from Mini 2 is conducted in the outdoor environment (Figure 3.1) in New Richmond, IN.

Figure 3.12 shows the DJI Mavic 2 Pro [183]. Mavic 2 Pro weighs about 0.907 kg, and the diagonal length of the wheelbase is about 354.0 mm. The experiment of collecting audio data from Mavic 2 Pro is conducted in the outdoor environment (Figure 3.1) in New Richmond, IN.

Figure 3.11. DJI Mini 2

Figure 3.12. DJI Mavic 2 Pro

Figure 3.12 shows the DJI Air 2s [184]. Air 2s weighs about 0.595 kg, and the diagonal length of the wheelbase is about 302.0 mm. The experiment of collecting audio data from Air 2s is conducted in the outdoor environment (Figure 3.1) in New Richmond, IN.

Figure 3.14 shows the DJI Phantom 2 [185]. Phantom 2 weighs about 1.000 kg with batteries and propellers installed, and the diagonal length of the wheelbase is about 350.0 mm. The experiment of collecting audio data from Phantom 2 is conducted in the outdoor environment (Figure 3.1) in New Richmond, IN.

Figure 3.13. DJI Air 2s

Figure 3.14. DJI Phantom 2

Figure 3.15 shows the DJI Phantom 4 [186]. Phantom 4 weighs about 1.380 kg with batteries and propellers installed, and the diagonal length of the wheelbase is about 350.0 mm. The experiment of collecting audio data from Phantom 4 is conducted in the outdoor environment (Figure 3.1) in New Richmond, IN.

Figure 3.16 shows the DJI RoboMaster TT Tello Talent [187]. Tello Talent weighs about 0.087 kg with batteries and propellers installed. The dimensions are 98.0*92.5*41.0 mm. RoboMaster TT is based on open source with a built-in ESP32 chip. Tello TT is developed and improved based on the previous model Tello EDU. The experiment of collecting audio

Figure 3.15. DJI Phantom 4

data from Tello Talent is conducted in the indoor environment in K-SW at Purdue University West Lafayette, IN.

Figure 3.16. DJI RoboMaster TT Tello Talent

Figure 3.17 shows the Hasakee Q11 [188]. Q11 weighs about 0.010 kg with batteries installed. The diagonal length of the wheelbase is about 228.6 mm. The experiment of collecting audio data from Q11 is conducted in the indoor environment in K-SW at Purdue University West Lafayette, IN.

Figure 3.18 shows the Syma X5SW [189]. X5SW weighs about 0.119 kg with batteries installed, and the dimensions are 315.0*315.0*105.0 mm. The experiment of collecting audio

Figure 3.17. Hasakee Q11

data from X5SW is conducted in the indoor environment in K-SW at Purdue University West Lafayette, IN.

Figure 3.18. Syma X5SW

Figure 3.19 shows the Syma X5UW [189]. X5UW weighs about 0.127 kg with batteries installed, and the diagonal length of the wheelbase is about 368.3 mm. The experiment of collecting audio data from X5UW is conducted in the indoor environment in K-SW at Purdue University West Lafayette, IN.

Figure 3.20 shows the Syma X20 [190]. X20 weighs about 0.181 kg with a battery, and the diagonal length of the wheelbase is about 127 mm. The experiment of collecting audio

Figure 3.19. Syma X5UW

data from X20 is conducted in the indoor environment in K-SW at Purdue University West Lafayette, IN.

Figure 3.20. Syma X20

Figure 3.21 shows the Syma X20P [191]. X20P weighs about 0.181 kg with battery, and the dimensions are 105.0*105.0*25.0 mm. The experiment of collecting audio data from X20P is conducted in the indoor environment in K-SW at Purdue University West Lafayette, IN.

Figure 3.22 shows the Syma X26 [192]. X26 weighs about 0.450 kg with a battery, and the dimensions are 44.0*131.0mm*131.0 mm. The experiment of collecting audio data from

Figure 3.21. Syma X20P

X26 is conducted in the indoor environment in K-SW at Purdue University West Lafayette, IN.

Figure 3.22. Syma X26

Figure 3.23 shows the UDI U46 [193]. U46 weighs about 0.200 kg with battery, and the dimensions are 91.4*81.2*33.0 mm. The experiment of collecting audio data from U46 is conducted in the indoor environment in K-SW at Purdue University West Lafayette, IN.

Figure 3.23. UDI U46

3.3 Overall System

Figure 3.24 presents the overall process for the proposed UAV audio classification system. Firstly, the audio data is recorded and labeled with UAV manufacture, model, and data/time information. Then, feature extraction is performed to obtain MFCCs from an audio sample, firstly, the audio file is converted from a time domain signal to a frequency domain signal by using Discrete Fourier Transform. Secondly, we calculate the logarithm of the Mel representation of audio, followed by taking logarithmic magnitude and applying discrete cosine transformation. From the above steps, a cepstrum feature created from Mel frequencies is the MFCC feature. The next step is to evaluate and verify the dataset by training different machine learning models, including Gradient Boosting, KNN, Decision Tree, Random Forest, and Gaussian Naive Bayes. After evaluating the dataset, the extracted features then are used to feed into the CNN model to train a UAV classification system. After the training, an evaluation is performed to test how the system reacts to the data it never saw before. Lastly, the model is assessed with accuracy and loss in training and testing and confusion matrix.

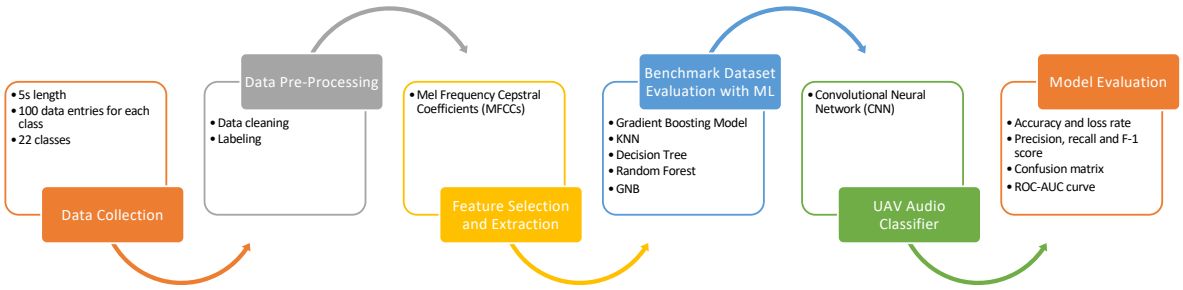


Figure 3.24. Audio-based UAV Classification System Overview

3.4 Audio Feature Extraction

Similar to common practice in audio-based ML, instead of feeding a whole audio dataset to ML models, I use the compact representation of the audio signal, instead of the raw audio files. With that, the proposed system reduces the form of feature representation in both size and dimensions of the audio data. In this research, MFCCs are chosen (mel frequency cepstral coefficients) as the feature extraction method. MFCC is one of the most widely used feature in audio processing applications such as surveillance-related events[1], environmental sound classification[118], speech recognition[119], speech enhancement [120], speaker recognition[121], music genre classification[122], and more.

When extracting MFCC features from audio data, 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:

$$\hat{x}(k) = \sum_{n=0}^{N-1} x(n) exp(-\mathrm{i}2\pi \frac{kn}{N}) \tag{3.1}$$

where k is the index of DFT output in the frequency domain. And $k = -\frac{N}{2}, ..., -1, 0, 1, ..., \frac{N}{2} - 1$. $\hat{x}(k)$ 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 Cosine Transform (DCT) is done using a Fast Fourier transform. The power spectrum $ps(f)$ is defined as follows [194]:

$$ps(f) = \sqrt{Re(H_f)^2 + Im(H_f)^2} \tag{3.2}$$

The next step is to apply Mel filter banks to the power spectrum using the Mel scale and the result is called mel (melody) spectrogram. Mel scale is defined as follows, where $m(f)$ indicates the frequencies on the mel scale measured in mels and f is the normal frequencies measured in Hz:

$$m(f) = 2595 \cdot \log_{10}(1 + \frac{f}{700}) \tag{3.3}$$

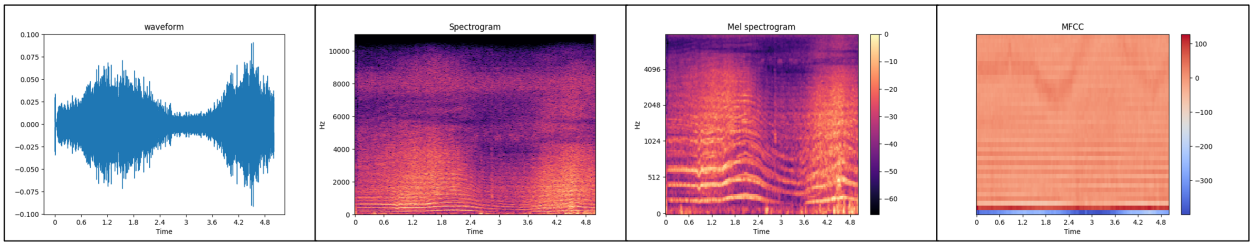
The number of filters is 2040 and 26 is a standard number to use. These 26 numbers indicate the energy in each filter bank. The purpose of applying mel scale is that to mimic how human hearing precepts sound. The name mel comes from the word 'melody".

The final step of extracting MFCCs is to calculate the cepstral coefficients based on the previous step, using Discrete Cosine Transformation (DCT) 26 log filter bank energies. The calculation of cepstrum is defined as following [194], where $x(t)$ is the time-domain signal:

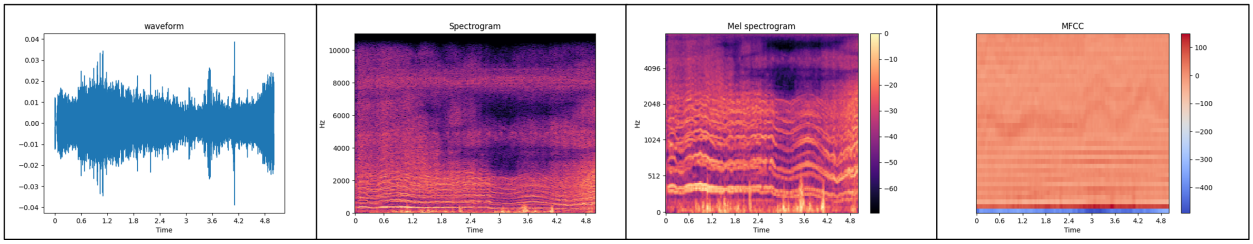
$$c_d = \frac{1}{M} \sum_{m=0}^{M-1} C_m cos(\frac{\pi(2d+1)m}{2M}) \tag{3.4}$$

In the above formula, c_d is the d^th cepstral coefficient, and M is the total number of filter banks. C_m indicates the log energy for filter bank m. Typically, c_1 c_12 constitute the MFCCs. The above steps result in MFCC features, which is a Mel spectrum opposed to time.

Figure 3.25 is a comparison of the MFCC extraction process with two UAVs: DJI Mavic Air 2 and DJI Mini 2. These two UAVs are similar in design and appearance, but Air 2 is



(a) DJI Mavic Air 2



(b) DJI Mini 2

Figure 3.25. MFCC Features Extraction Process.

about 330 grams heavier than Mini2. The very first image to the left in the process is the visual representation of the waveform from the audio data. The x-axis is time, and the y-axis is amplitude. The second to the left is the spectrogram using DFT, in which the x-axis is time and the y-axis is frequency. The next image is the Mel Spectrogram, based on the mel scale. The last step image the final results of MFCC features. From each step, we can see the differences in visual representations of the two UAVs.

3.5 Choice of Neural Network

For the effectiveness of neural networks in processing a large amount of data, it has been one of the widely adopted techniques for audio-based research and applications[195]. Traditional machine learning models such as linear regression and K-nearest neighbors can also be applied to audio classification. However, considering the scale of our dataset, which contains more than 20 different types of UAVs, and the model complexity of neural networks, which often surpass traditional machine learning algorithms, I decide to use neural networks as the training model to perform audio classification of UAVs.

Frequently used neural networks include, but are not limited to, feed-forward neural networks, Convolutional Neural Networks (CNNs), Multi-Layer Perceptron (MLP), and Recurrent Neural Networks (RNNs). Among these models, this study uses CNN to perform experiments. CNN is a feed-forward network that contains convolution layers and is specifically designed to process data with grid-like topologies in images[17]. In current and previous studies, along with the use of computer vision techniques, CNN produces some of the best results in the image processing areas, such as object detection, image classification, image segmentation, etc [18]. Recent works in the audio area prove that CNNs are also effective with audio data, and have been used in different audio-related tasks including audio classification[195].

When comparing other neural networks, CNN can achieve a similar performance while requiring less memory by reducing the number of parameters because of two reasons: local receptive fields and weight sharing. CNN usually consists of multiple convolutional layers with one or more dense layers existing afterward. In our study, the proposed CNN model contains two convolutional layers, two dense layers, and one dropout layer.

Figure 3.26 shows the proposed CNN structure in the first phase of the experiment. The model first contains a 1D convolutional layer with 16 filters as the input layer. Then, a second 1D convolutional layer with 4 filters is added as the first hidden layer. After each 1D convolutional layer, a ReLU activation layer is added to introduce the non-linearity for classifying multiple types of UAVs. The convolution kernel size is set to 3 for both convolutional layers. After two layers of convolution, the model flattens the output from the convolutional part of the network using a flatten layer. The flatten layer takes in the previous convolutional layers multi-dimensional output and transforms it into one-dimensional. Then, one dense layer is added to connect all the neurons for computing, and the dense layer outputs shape is a one-dimensional array, which includes 32 elements. ReLU activation function is also applied to this layers output. To overcome overfitting and improve model generalization, one dropout layer is added before the output layer. Moreover, the dropout rate is set to 0.4. Finally, a dense layer with softmax activation that has the same number of neurons/nodes as the number of classes to be classified is added as the output layer to make the multi-class

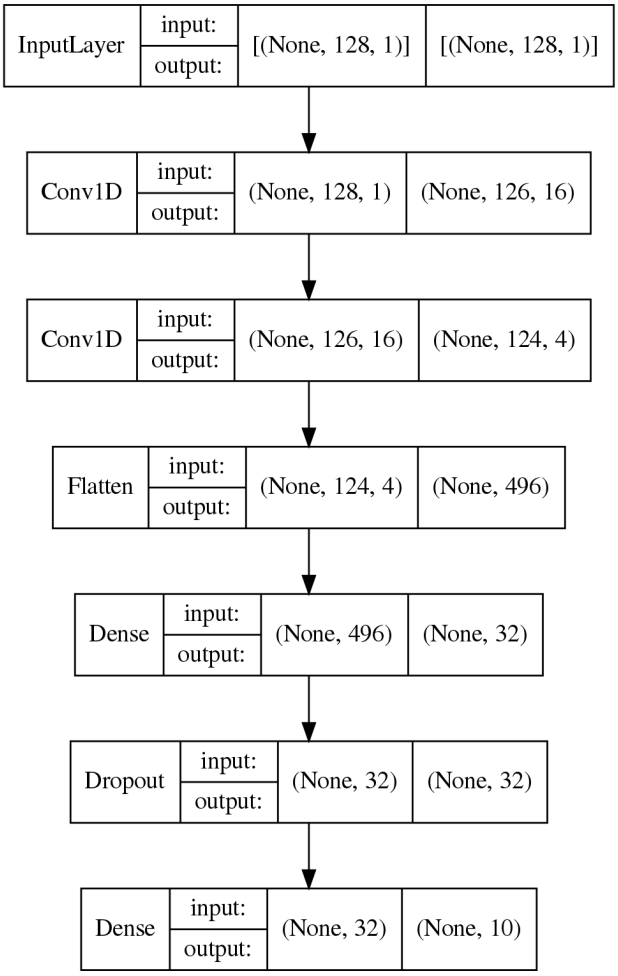


Figure 3.26. CNN Structure for 10 Classes

classification. Our initial dataset contains 10 categories of UAVs, therefore, the output layer is a 10 nodes softmax layer.

3.6 Summary

This chapter first presents the process of data collection 3.1. Data collection is divided into indoor and outdoor, depending on the UAVs' availability. 14 out of 22 UAVs are flown and the data is collected outdoors, and the location is shown in Figure 3.1. The rest 8 UAVs are flown and data are collected indoors in K-SW at Purdue University, West Lafayette, IN. Section 3.2 provides detailed specifications of UAVs type, weight, size, and more. Section 3.3 is the system overview for the proposed UAV classifier using audio data. Section 3.4 provides a detailed explanation of how feature extraction works, and how to calculate MFCCs. Lastly, Section 3.4 presents the proposed CNN structure for phase one. However, phase two and three adopted a similar structure to the CNN, which is used in the first and second phase. One more dropout layer is added between the two convolutional layers in phase three to prevent overfitting. The details can be found in the next chapter.

4. EXPERIMENT

4.1 First Phase of the Experiment

4.1.1 UAV Audio Dataset

In the first stage, 10 different UAVs are used and 1043 audio files are collected, each containing 5 seconds of recording of the flying drone audio data [196], as shown in Table 4.1. The entire dataset is 1.8GB in size. The manufacturers of the UAVs include DJI, Autel, Syma, and Yuneec. I operate and record UAVs by DJI, Autel Robotics, and Yuneec in the outdoor environment, and UAVs by Syma in the indoor environment. Syma UAVs could not withstand well outdoors and usually drifted away in the wind. The outdoor operation of drones takes place in an open area in New Richmond, Indiana, and the coordinate is (40.2227062, -87.0000169). The indoor recordings are conducted in K-SW at Purdue University.

4.1.2 Data Collection

UAVs used in the first phase of the experiment include five UAVs from DJI, three UAVs from Syma, one from Autel, and one from Yuneec. I use a MacBook Air to record all the audio data. The MacBook Air has a 1.1GHz Quad-Core Intel Core i5 CPU and 8GB of memory. Initially, I collect audio samples with different duration, which are 3s, 5s, and 10s. I trained a basic CNN model with one convolutional layer and a KNN (K-Nearest Neighbors) model using different lengths of audio data. The result indicates that 5s and 10s data samples produced similar accuracy. Hence, I continued the data collection process with a length of 5s for each audio file.

For indoor data collection, the Syma drones are used to fly around the room, with different movements, including hovering, ascending, and descending, rolling left or right, pitching forwards or backward, and rotating left or right. The laptop is placed about 30 inches from the ground on a table in the center of the room.

Besides, the DJI drones, Autel Evo, and the Yuneec Typhoon are used for outdoor audio data collection. The maximum height is about 20 meters, and the maximum radius from the recording station is about 20 meters. Audio data is recorded when drones were flying

Table 4.1. UAV Audio Dataset 10 Classes

Manufacture	Model	Number of Files	Duration
DJI	Matrice 200	100	500
DJI	Matrice 200 V2	105	525
DJI	Mavic 2 Pro	100	500
DJI	Phantom 2	106	530
DJI	Phantom 4	100	500
Autel	Evo 2 Pro	100	500
Syma	X5SW	110	550
Syma	X5UW	105	525
Syma	X20P	104	520
Yuneec	Typhoon H Plus	113	565
Total	-	1043	5215 (sec)

Table 4.2. Benchmark Evaluation Results with ML Models for 10 Classes

ML Model	Accuracy
Gradient Boosting Training	0.923
Gaussian Naive Bayes	0.92
Decision Tree	0.79
Random Forest	0.94
K-Nearest Neighbour	0.89

with different movements, including hovering, ascending, and descending, rolling left or right, pitching forwards or backward, and rotating left or right. The laptop is placed on a table facing the open field. The height of the table is about 30 inches. No extra filtering is processed on the audio data. Hence, the sound of the wind, birds chirping, and some traffic noise are also included in the recordings. Also, the weather condition during the outdoor recording varied, which include sunny, cloudy, and foggy days. The wind speed is in the range of 3mph to 13mph. The temperatures are between 65 to 90 degrees Fahrenheit. The relative humidity for the days to collect data is between 73% and 84%.

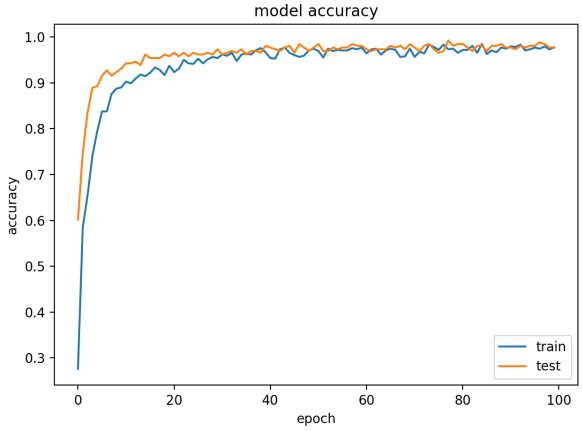
4.1.3 Dataset Evaluation for 10 Classes

The 10-class dataset is evaluated by the results from five different ML models trained with the collected dataset. The selected ML models are Gradient Boosting, Gaussian Naive Bayes,

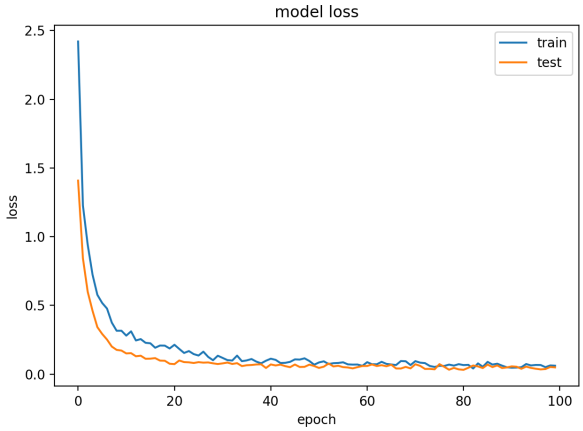
Decision Tree, Random Forest, and K-Nearest Neighbour. Table 4.2 shows the accuracy score from each model. The dataset is split into 75% training and 25% testing. For KNN, I choose 3 as the K value in training. I only use the default values for hyperparameters for all the other ML models. Three out of five models achieve accuracy scores above 90%. Among them, Random forest produces the highest accuracy of 94%. Decision Trees has the lowest of 79%. These results provide a baseline for how well the ML model performs, which is trained with the benchmark dataset.

4.1.4 Convolutional Neural Network Training

To train the proposed CNN model, as shown in Figure 3.26, the dataset is split into training and testing sets. 75% data are included in the training set while the rest 25% data are used for testing. As a result, the training dataset contains 782 recordings and the test set has 261 recordings. I choose Adam as the optimizer in training and the learning rate is set to 0.001. Categorical cross-entropy is selected for the loss function and the model is trained with batch size 32. The model is trained on a server with an Intel Core i7-5930K CPU, 64GB of memory, and an Nvidia GeForce RTX 2070 Super 8GB GPU. The server is installed with



(a) Training and Testing Accuracy Plot for 10 Classes



(b) Training and Testing Loss Plot for 10 Classes

Figure 4.1. Evaluation Results for 10 Classes

Tensorflow version 2.8.0 combined with Python 3.8.14 as the framework to implement the CNN model.

The model is trained for 100 epochs and the average training time using the above configuration is 17.31 seconds with a standard deviation of 0.44 seconds. This result is collected from 10 i.i.d model training runs. The model accuracy history and loss history plots are shown in Figure 4.1a and Figure 4.1b, respectively. In both plots, the line shown in blue is the training accuracy and loss history, and the yellow line shows the test accuracy and loss history. As shown in the history plots, the model converges after epoch 60 since the accuracy is not increasing as well as the loss is not decreasing as the training continues. Also, we can see from the plots that the accuracy and loss of the training dataset and testing dataset are not parting away from each other after the model saturates. Therefore, no overfitting nor underfitting is happening when training the model with 100 epochs.

Table 4.3. Accuracy, Precision, Recall, and F1-scores for 10 Classes Dataset

UAV Model	Precision	Recall	F1 Score	Support
DJI Matrice200	1.00	1.00	1.00	25
DJI Matrice200 V2	1.00	1.00	1.00	28
DJI Mavic2_Pro	1.00	0.96	0.98	26
DJI Phantom2	1.00	1.00	1.00	25
DJI Phantom4	0.83	1.00	0.91	25
Autel EvoII_Pro	0.96	0.83	0.89	29
Syma X5SW	1.00	1.00	1.00	23
Syma X5UW	1.00	1.00	1.00	21
Syma X20P	1.00	1.00	1.00	24
Typhoon H Plus	1.00	1.00	1.00	35

4.1.5 Results Evaluation and Analysis

After the training is finished, I evaluate the trained model using the classification results from the test dataset. The test dataset is tested using the trained CNN model and the precision, recall, and F1 score of each category are calculated. Table 4.3 shows the calculated numbers of each score as well as the number of recordings used (column support) for all types of UAVs. As the table shows, the precision of each type of UAV's classification result is

Table 4.4. UAV Audio Dataset 15 Classes

Manufacture	**Model**	**Number of Files**	**Duration**
David	Tricopter	102	510
DJI	Matrice200	100	500
DJI	Matrice200 V2	105	525
DJI	Mavic Air 2	131	655
DJI	Mavic Mini 1	120	600
DJI	Mini 2	116	580
DJI	Mavic 2 Pro	100	500
DJI	Phantom 2	106	530
DJI	Phantom 4	100	500
Autel	Evo 2 Pro	100	500
Syma	X5SW	110	550
Syma	X5UW	105	525
Syma	X20	112	560
Syma	X20P	104	520
Yuneec	Typhoon H Plus	113	565
Total	-	1624	8120 (sec)

above 83%, from which 8 UAV models' classification results reach a precision score of 100%. Furthermore, all recall values are also above 83% with 8 of them equal to 100%. The high precision values and high recall values indicate the model is a good classifier where not only does the model's classification result have a low number of false positives, but also has a low false negative rate. What's more, the calculated F1 scores further prove that the proposed CNN model has great performance when classifying UAVs using audio data.

The overall test accuracy of the trained model is 97.7% and the test loss is 0.049. I trained the same model 10 times using the proposed dataset and collected the test accuracy and test loss results. The average test accuracy of 10 separate trained models is 97.43% and the standard deviation is 0.9%, and the average test loss is 0.085 with a standard deviation of 0.051. The statistics show that the proposed CNN model and the training configuration have a persistent overall performance and can be reproduced for further research.

Table 4.5. Benchmark Evaluation Results with ML Models for 15 Classes

ML Model	Accuracy
Gradient Boosting Training	0.94
Gaussian Naive Bayes	0.92
Decision Tree	0.79
Random Forest	0.95
K-Nearest Neighbour	0.89

4.2 Second Phase of the Experiment

I am planning to continue collecting more data, and eventually have at least 20 different classes in the UAV audio dataset. In the meantime, evaluate the dataset with different ML models. The CNN model needs to be tuned with more classes to achieve the expected classification results.

4.2.1 Data Collection

In the second stage, I used 15 different UAVs and collected 1624 audio files, each containing 5 seconds of recording of the flying drone audio data, as shown in Table 4.4. The entire dataset is 2.7GB in size. In addition to the dataset collected in the first phase, 3 more DJI drones are recorded outside on the same farm as the rest of the drones, 1 Syma drone is recorded in K-SW at Purdue University, and a tricopter built by a friend of mine is recorded in Columbus, IN. All the data collection methods are the same as in the first phase. The wind speed is between 5mph to 13mph on the days of data collecting. The temperatures are between 39 and 79 degrees Fahrenheit. The average humidity is around 64%.

4.2.2 Dataset Evaluation

The 15-class dataset is evaluated by the results from five different ML models trained with the collected dataset. The selected ML models are Gradient Boosting, Gaussian Naive Bayes, Decision Tree, Random Forest, and K-Nearest Neighbour. Figure 4.3 shows the accuracy score from each model. The dataset is split into 75% training and 25% testing. For KNN, I

choose 3 as the K value in training. I only use the default values for hyperparameters for all the other ML models. Three out of five models achieve accuracy scores above 90%. Among them, Random forest produces the highest accuracy of 95%. Decision Trees has the lowest of 79%. These results provide a baseline for how well the ML model performs, which is trained with the benchmark dataset.

4.2.3 Convolutional Neural Network Training

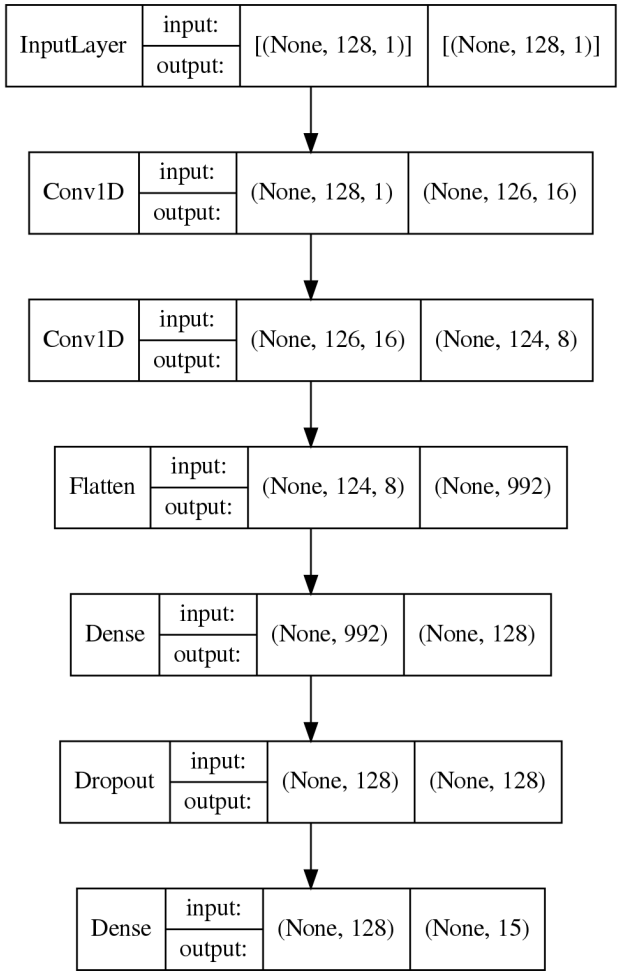


Figure 4.2. CNN Structure for 15 Classes

Figure 4.2 illustrates the structure of CNN in the second phase. Compared to the one in the first phase, the number of filters is increased in the second convolution layer from 4 to 8. I also adjust the number of elements in the dense layer after the two convolution layers,

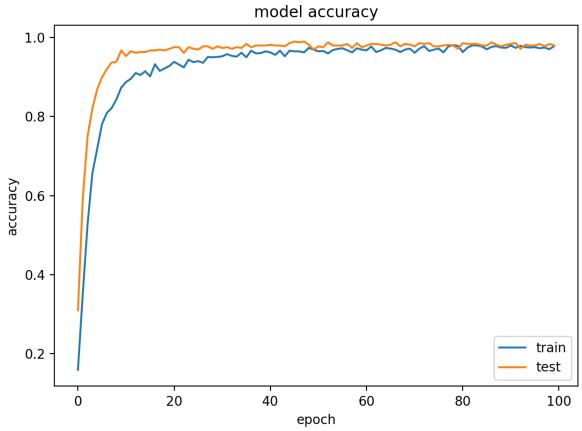
from 32 to 128. Finally, the last dense layer has the same number of neurons as the number of classes to be classified as the output layer to perform multi-class classification tasks. In the second phase, the dataset contains 15 categories of UAVs, thus, the output layer is a 15 nodes softmax layer.

To train the proposed CNN model, the dataset is split into training and testing sets. 75% data are included in the training set while the rest 25% data are used for testing. As a result, the training dataset has 1218 recordings included in the training set and 406 for the test set. The training uses Adam as the optimizer and the learning rate is set to 0.001. Categorical cross-entropy is used as the loss function and trained the model with batch size 32. The model is trained on the same server used for the first phase experiment. Tensorflow version 2.8.0 and Python 3.8.14 are used as the framework to implement the CNN model.

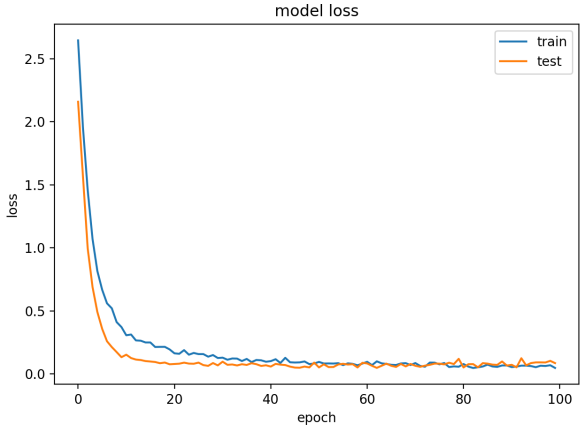
I train the model for 100 epochs and the average training time using the above configuration is 22.39 seconds with a standard deviation of 0.005 seconds. This result is collected from 10 i.i.d model training runs. The model accuracy history and loss history plots are shown in Figure 4.3a and Figure 4.3b, respectively. In both plots, the line shown in blue is the training accuracy and loss history, and the yellow line shows the test accuracy and loss history. As shown in the history plots, the model converges after epoch 40 since the accuracy is not increasing as well as the loss is not decreasing as the training continues. Also, we can see from the plots that the accuracy and loss of the training dataset and testing dataset are not parting away from each other after the model saturates. Therefore, no overfitting nor underfitting is happening when training the model with 100 epochs.

4.2.4 Result Evaluation and Analysis

Same as the first stage experiment, I evaluate the trained model using the classification results from the test dataset. The test dataset is classified using the trained CNN model and precision, recall, and F1 score of each category are calculated. Table 4.6 illustrates the calculated number of each score and the number of recordings used for all UAVs. We can see that the precision of each type of UAV's classification is above 88%, with 11 of them achieving 100%. In addition, all the recall scores are above 88%, and also 11 of them are



(a) Training and Testing Accuracy Plot for 15 Classes



(b) Training and Testing Loss Plot for 15 Classes

Figure 4.3. Evaluation Results for 15 Classes

Table 4.6. Accuracy, Precision, Recall, and F1-scores for 15 Classes Dataset

UAV Model	Precision	Recall	F1 Score	Support
David Tricopter	1.00	1.00	1.00	23
DJI Matrice 200	0.96	1.00	0.98	26
DJI Matrice 200 V2	0.96	1.00	0.98	22
DJI Mavic Air 2	1.00	0.88	0.94	33
DJI Mavic Mini 1	1.00	1.00	1.00	27
DJI Mini 2	1.00	1.00	1.00	24
DJI Mavic 2 Pro	1.00	1.00	1.00	29
DJI Phantom 2	1.00	1.00	1.00	32
DJI Phantom 4	1.00	1.00	1.00	25
Autel Evo 2 Pro	0.93	1.00	0.96	26
Syma X5SW	0.88	1.00	0.94	23
Syma X5UW	1.00	0.96	0.98	28
Syma X20	1.00	0.96	0.98	24
Syma X20P	1.00	0.94	0.97	32
Yuneec Typhoon H Plus	1.00	1.00	1.00	32

100%. Again, similar to the first stage experiment, the high precision values and high recall values indicate our model is a good classifier where not only does the model's classification results have a low number of false positives and false negatives. Furthermore, the calculated F1 scores indicate that the proposed CNN model has great performance when classifying UAVs using audio data.

The overall test accuracy of the trained model is 98.7% and the test loss is 0.076. I train the same model 10 times using the proposed dataset. The average test accuracy of 10 separate trained models is 98.2% and the standard deviation is 0.9%, and the average test loss is 0.085 with a standard deviation of 0.0052. The statistics indicate that the proposed CNN model and the training configuration have a persistent overall performance and can be reproduced for further research.

Table 4.7. UAV Audio Dataset 22 Classes

Manufacture	**Model**	**Number of Files**	**Duration**
Self-build	David Tricopter	102	510
Self-build	PhenoBee	116	585
Autel	Evo 2 Pro	100	500
DJI	Matrice 200	100	500
DJI	Matrice 200 V2	105	525
DJI	Mavic Air 2	131	655
DJI	Mavic Mini 1	120	600
DJI	Mini 2	116	580
DJI	Mavic 2 Pro	100	500
DJI	Mavic 2s	115	580
DJI	Phantom 2	106	530
DJI	Phantom 4	100	500
DJI	RoboMaster TT Tello	118	595
Hasakee	Q11	108	545
Syma	X5SW	110	550
Syma	X5UW	105	525
Syma	X20	112	560
Syma	X20P	104	520
Syma	X26	138	695
Swellpro	Splash 3 plus	120	605
Yuneec	Typhoon H Plus	113	8120
UDI RC	U46	101	510
Total	-	2440	12200 (sec)

4.3 Third Phase of the Experiment

4.3.1 Data Collection

In the third stage, I used 22 different UAVs and collected 2440 audio files, each containing 5 seconds of recording of the flying drone audio data, as shown in Table 4.7. The entire dataset is 6GB in size. In addition to the dataset collected in the second phase, DJI Mavic Mini2, and Swellpro Splash 3 plus are recorded outside in the same location as the other outdoor ones. Syma X26, Hasakee Q11, UDIRC U46 are recorded in K-SW at Purdue University. We also included two self-build drones: 1 tricopter built by a friend of mine is recorded in Columbus, IN, and an agriculture UAV is recorded near Agronomy Center for Research and Education in West Lafayette, IN. All the data collection methods are the same as in the first and second phases. The wind speed is between 5mph to 20mph on the days of data collecting. The temperatures are between 30 and 79 degrees Fahrenheit. The average humidity is around 72%.

4.3.2 Dataset Evaluation for 22 Classes

The 22-class dataset is evaluated by the results from five different ML models trained with the collected dataset. The selected ML models are Gradient Boosting, Gaussian Naive Bayes, Decision Tree, Random Forest, and K-Nearest Neighbour. Figure 4.5 shows the accuracy score from each model. The dataset is split into 80% training and 20% testing. For KNN, the K value is set to 3 in training. I only use the default values for hyperparameters for all the other ML models. Four out of five models achieve accuracy scores above 94%. Among them, Random Forest performs the best with the highest accuracy of 97.5%. Decision Trees has the lowest of 82.8%. These results provide a baseline for how well the ML model performs, which is trained with the benchmark dataset.

4.3.3 Convolutional Nerual Network Training

Figure 4.4 illustrates the structure of CNN in the third phase. Compared to the one in the second phase, the number of filters is increased in both convolution layers to 32. The

Table 4.8. Benchmark Evaluation Results with ML Models for 22 Classes

ML Model	Accuracy
Gradient Boosting Training	0.945
Gaussian Naive Bayes	0.957
Decision Tree	0.828
Random Forest	0.975
K-Nearest Neighbour	0.941

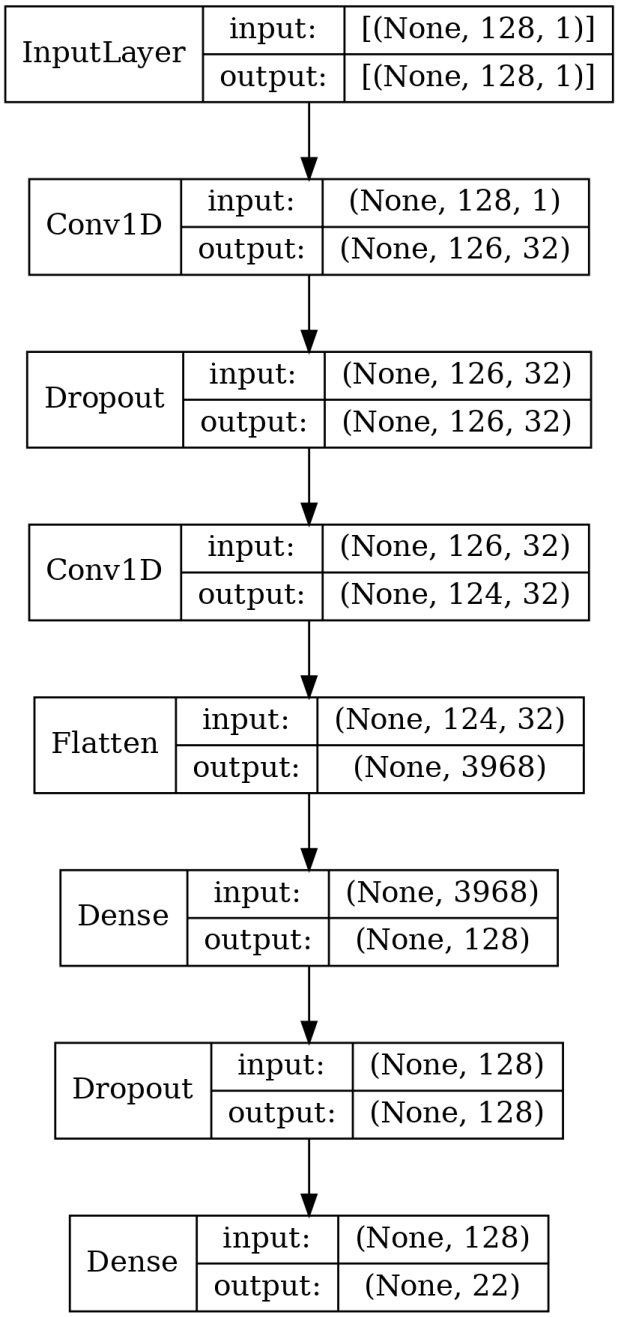
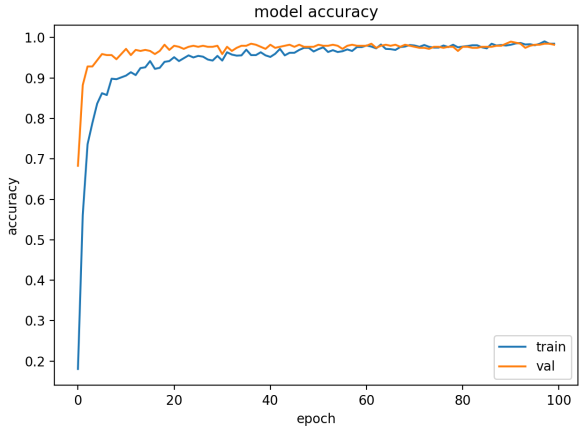


Figure 4.4. CNN Structure for 22 Classes

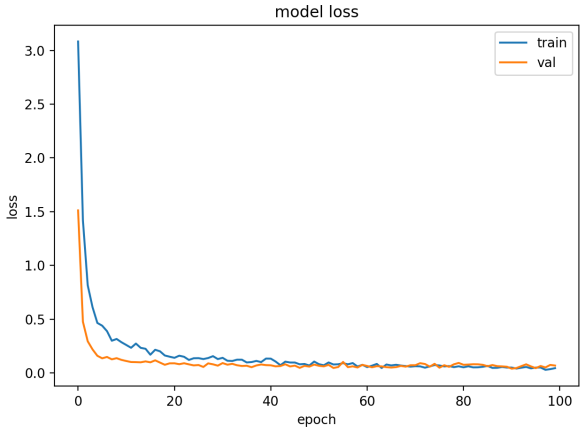
number of elements in the dense layer after the two convolution layers are the same as in the second stage, which is 128. Finally, the last dense layer has the same number of neurons as the number of classes to be classified as the output layer to perform multi-class classification tasks. In the third phase, the dataset contains 22 categories of UAVs, thus, the output layer is a 22 nodes softmax layer. I also added another drop out layer in between the two convolution layers to prevent overfitting.

To train the proposed CNN model, the dataset is split into training and testing sets. 80% data are included in the training set while the rest 20% data are used for testing. As a result, the training dataset has 1952 recordings included in the training set and 488 for the test set. The training uses Adam as the optimizer and the learning rate is set to 0.001. Categorical cross-entropy is used as the loss function and trained the model with batch size 32. The model is trained on the same server used for the first phase experiment. Tensorflow version 2.8.0 and Python 3.8.14 are used as the framework to implement the CNN model.

I train the model for 100 epochs and the average training time using the above configuration is 41.1 seconds with a standard deviation of 0.467 seconds. This result is collected from 10 i.i.d model training runs. The model accuracy history and loss history plots for the $10th$ trained model are shown in Figure 4.5a and Figure 4.5b, respectively. In both plots, the line shown in blue is the training accuracy and loss history, and the yellow line shows the test accuracy and loss history. As shown in the history plots, the model converges after epoch 60 since the accuracy is not increasing as well as the loss is not decreasing as the training continues. Also, we can see from the plots that the accuracy and loss of the training dataset and testing dataset are not parting away from each other after the model saturates. Therefore, no overfitting nor underfitting is happening when training the model with 100 epochs. In addition, average accuracy scores from 10 training on the test set for each individual class are presented in Figure 4.6. The lowest accuracy score belongs to the class of Phenobee, which is 0.9465%.



(a) Training and Testing Accuracy Plot for 22 Classes



(b) Training and Testing Loss Plot for 22 Classes

Figure 4.5. Evaluation Results for 22 Classes

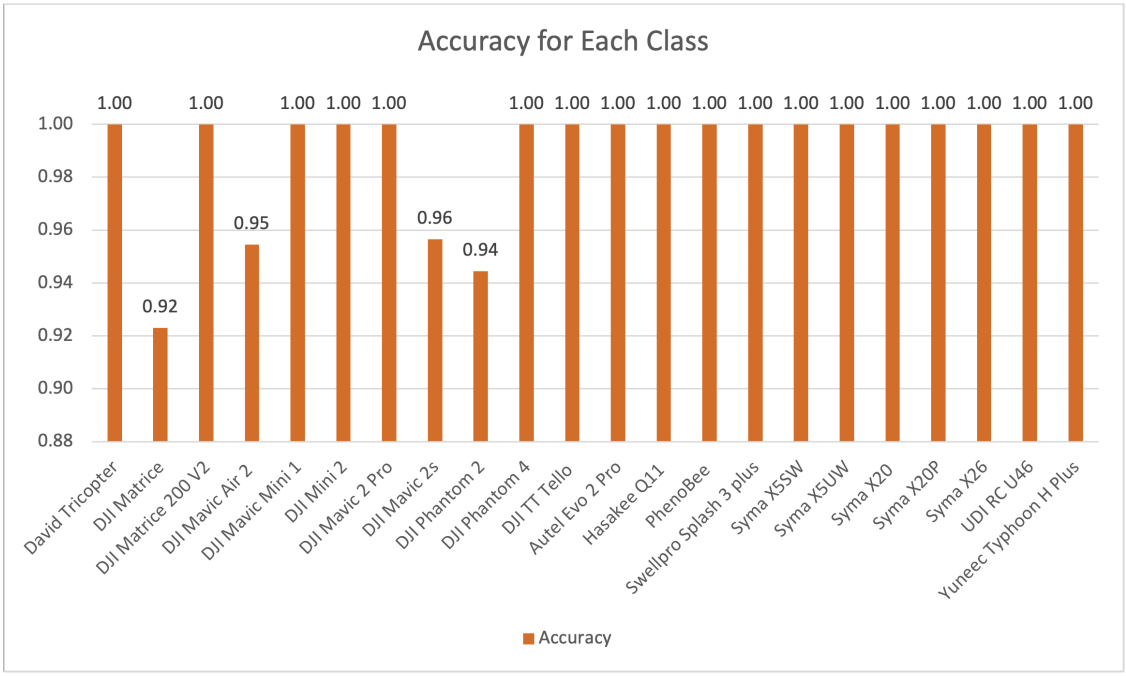


Figure 4.6. Accuracy Score for Each Class

4.3.4 Result Evaluation and Analysis

I train the same model 10 times using the proposed dataset. The average test accuracy of 10 separate trained models is 99.1% and the standard deviation is 0.38%, and the average test loss is 0.027 with a standard deviation of 0.8%. The overall test accuracy of the $10th$ trained model is 99.1% and the test loss is 0.007. The statistics indicate that the proposed CNN model and the training configuration have a persistent overall performance and can be reproduced for further research.

Table 4.9. Accuracy, Precision, Recall, and F1-scores for 22 Classes Dataset

UAV Model	**Label**	**Precision**	**Recall**	**F1 Score**	**Support**
David Tricopter	0	1.00	1.00	1.00	22
DJI Matrice 200	1	1.00	0.92	0.96	13
DJI Matrice 200 V2	2	0.94	1.00	0.97	15
DJI Mavic Air 2	3	1.00	0.95	0.98	22
DJI Mavic Mini 1	4	1.00	1.00	1.00	21
DJI Mini 2	5	1.00	1.00	1.00	23
DJI Mavic 2 Pro	6	0.95	1.00	0.97	19
DJI Mavic 2s	7	0.96	0.96	0.96	23
DJI Phantom 2	8	1.00	0.94	0.97	18
DJI Phantom 4	9	1.00	1.00	1.00	25
DJI TT Tello	10	1.00	1.00	1.00	21
Autel Evo 2 Pro	11	1.00	1.00	1.00	18
Hasakee Q11	12	1.00	1.00	1.00	29
PhenoBee	13	0.96	1.00	0.98	23
Swellpro Splash 3 plus	14	1.00	1.00	1.00	24
Syma X5SW	15	1.00	1.00	1.00	21
Syma X5UW	16	1.00	1.00	1.00	27
Syma X20	17	1.00	1.00	1.00	19
Syma X20P	18	1.00	1.00	1.00	27
Syma X26	19	1.00	1.00	1.00	27
UDI RC U46	20	1.00	1.00	1.00	18
Yuneec Typhoon H Plus	21	1.00	1.00	1.00	33

Same as in the first and the second phase experiment, I evaluate the trained model using the classification results from the test dataset. The test dataset is classified using the trained CNN model and precision, recall, and F1 score of each category are calculated. Table 4.9 illustrates the calculated number of each score from the $10th$ training and the

number of recordings used for all UAVs. We can see that the precision of each type of UAV's classification is above 94%, with 18 of them achieving 100%. In addition, all the recall scores are above 92%, and also 18 of them are 100%. Again, even better than the first and the second stage experiment, the high precision values and high recall values indicate our model is a good classifier, for the model's classification results have a low number of false positives and false negatives. Furthermore, the calculated F1 scores indicate that the proposed CNN model has great performance when classifying UAVs using audio data.

The confusion matrix shown in Figure 4.7 is obtained by training the CNN classifier and evaluating the trained model on the test set. Let assign the name "CM" to the matrix, and each element in the matrix be indicated by "$CM\_truelabel, predictedlabel$," where "true label" is the row name which represents true labels), and "predicted label" is the column name that represents the predicted class. For example, $CM(Phantom4, Mavic2s) = 0$.

Figure 4.7 provides valuable information about the model performance as follows:

- The diagonal numbers present the data samples that are predicted correctly by the trained model. There are total of 484 data samples being correctly predicted among 488 test data samples, which results in 99.1% overall accuracy.

- $CM(Matrice200, X26) = 0$ indicates that the model does not confuse the data labeled as X26 with Matrice200. The classification model learned the similarities and differences between these two classes.

- $CM(Mavic\_Air2, Mavic2s) = 1$ implies that the model predicted one data sample that originally belonged to Mavic Air2 as Mavic2s. Thus, the accuracy in prediction for Mavic2s is 95.65%.

The other important tool to evaluate a classification model is to use AUC-ROC curves, which calculate the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC). The larger the area, the better the model. One vs rest (OvR) is one of the two methods when utilizing the ROC curve, and the other one is called One vs One (OvO). In this research, we only use OvR to evaluate the proposed classifier. OvR is often applied to evaluate multi-class classifiers by comparing each class against all the others at once. This

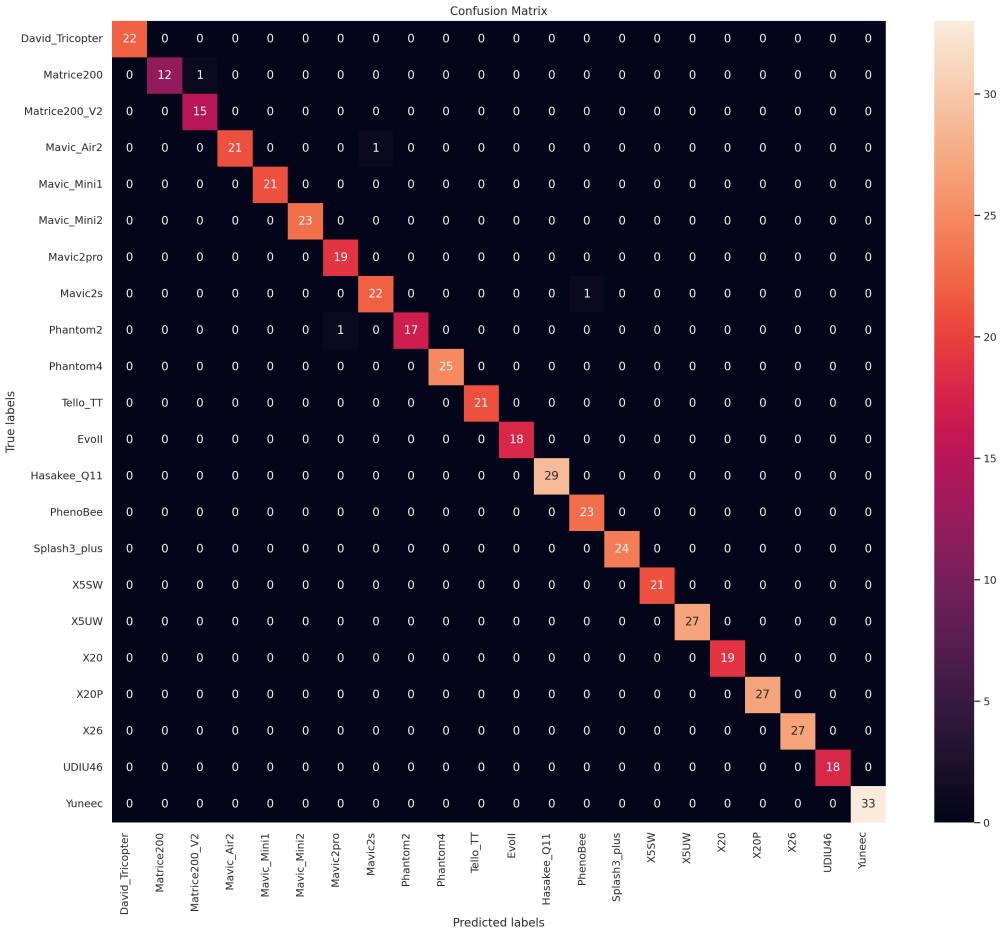


Figure 4.7. Confusion Matrix for 22 Classes

method takes one class to evaluate as the positive class, while all the rest of the other classes are the negative class. In this way, we simplify the output from the multi-class classification and convert it into a binary classification result. When analyzing the ROC curve, the diagonal dotted line indicates the random classifier, where the true positive rate and false positive rate are almost the same. Our classifier shows a high true positive and low false positive rate, as shown in Figure 4.8, which again indicates that the proposed CNN classifier has great performance when classifying UAVs using audio data. When plotting the ROC curve, numeric labels are used instead of UAV model names. Table 4.10 show the corresponding labels to the UAVs.

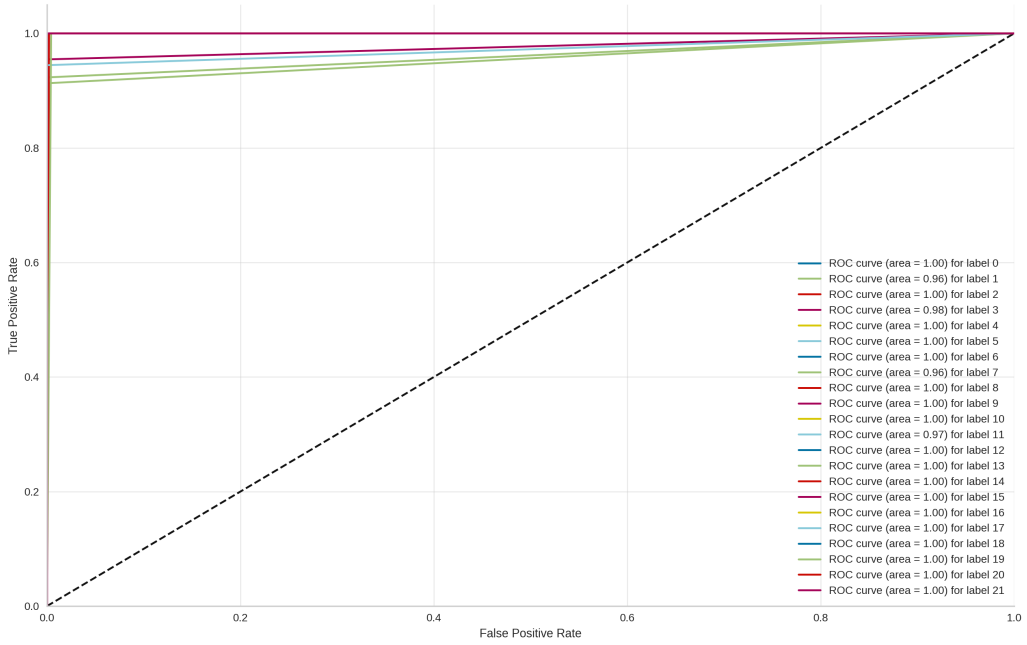


Figure 4.8. OvR ROC Curve

4.4 Summary

In this chapter, three phases of the experiments are presented. For each phase, a dataset evaluation is conducted, in which five Machine Learning models are trained using the collected

Table 4.10. UAV Models and Labels

0	David Tricopter	1	DJI Matrice 200	2	DJI Matrice 200 V2
3	DJI Mavic Air 2	4	DJI Mavic Mini 1	5	DJI Mini 2
6	DJI Mavic 2 Pro	7	DJI Mavic 2s	8	DJI Phantom 2
9	DJI Phantom 4	10	DJI TT Tello	11	Autel Evo 2 Pro
12	Hasakee Q11	13	PhenoBee	14	Swellpro Splash 3 plus
15	Syma X5SW	16	Syma X5UW	17	Syma X20
18	Syma X20P	19	Syma X26	20	UDI RC U46
21	Yuneec Typhoon H Plus				

dataset. All the ML models are trained with default settings and parameters. This step is to see how efficient the dataset is and to provide a baseline for Deep Learning model training. In the first phase (Section 4.1), audio data from 10 classes of UAVs are collected, and a CNN classifier is trained using the collected dataset. The average of 10 separate training is 97.43%, and the average test loss is 0.085. In the second phase of the experiment (Section 4.2), audio data from 15 classes of UAVs are collected and a CNN classifier is trained with the collected dataset. The CNN structure from the second phase is based on the first, but the number of filters is increased in both the convolutional layer and the dense layer. The average of 10 separate training is 98.2%, and the average test loss is 0.085. In the third phase (Section 4.3), audio data from 22 classes are collected and a CNN classifier is trained with the collected dataset. The CNN structure in the third phase has an increased number of filters in both convolutional layers, and another dropout layer is added between the two convolutional layers. The average of 10 separate training is 99.1%, and the average test loss is 0.027. Besides Accuracy, Precision, Recall, and F1-scores, additional evaluation metrics of the CNN structure in the third phase are conducted, including a confusion matrix plot Figure 4.7 and OvR ROC curves Figure 4.8.

5. CONCLUSION

The proliferation of unmanned aerial vehicles (UAVs), commonly known as drones, has experienced an exponential surge in popularity in recent years. These UAVs have become increasingly accessible to a wide range of users, encompassing both professionals and amateurs alike. However, the potential for malicious misuse of UAVs poses a significant threat to public safety. Presently, the regulation and enforcement of UAV guidelines primarily rely on self-regulation measures. Consequently, ensuring robust detection systems for UAVs has become paramount. Various methodologies, such as computer vision, radar, radio frequency, and audio-based approaches, have been employed to develop UAV detection systems. Nevertheless, each of these approaches possesses inherent advantages and disadvantages. In this research, the audio-based method is adopted due to its notable precision and cost-effectiveness, as it obviates the need for supplementary equipment. However, the paucity of publicly accessible datasets stands as a major impediment to the development of an audio-based UAV detection and classification system. To address this critical gap, the research is bifurcated into two principal components: firstly, the acquisition of a comprehensive and extensive UAV audio dataset, and secondly, the construction of a Deep Learning-based UAV classifier, which is trained utilizing the amassed dataset.

Chapter 1 provides the foundational background and rationale for this research (Section 1.1), which aims to address and mitigate the issue of malevolent unmanned aerial vehicle (UAV) threats and attacks through the utilization of UAV audio data and Deep Learning techniques. Moreover, Chapter 1 delves into the specifics of the research question (Section 1.2), highlighting its intrinsic importance and relevance within the field. Additionally, the chapter expounds upon the significance of this research (Section 1.3), shedding light on its potential implications and contributions. Furthermore, Chapter 1 acknowledges the limitations inherent in this study and outlines avenues for future exploration (Section 1.4).

Chapter 2 delves into significant architectures of Deep Learning, encompassing Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Seq2Seq models, and other relevant methodologies. Moreover, this chapter provides an in-depth exploration of

popular Deep Reinforcement Learning (DRL) algorithms pertaining to audio classification, namely model-based DRL, policy gradient-based DRL, and value-based DRL.

In light of the scarcity of publicly available audio datasets, Section **??** examines diverse approaches to data augmentation techniques. Additionally, a comprehensive literature review presented in Section 2.4 highlights the merits of utilizing Deep Learning (DL) in audio and speech processing applications. Specifically, a meticulous examination is conducted on ML-based UAV detection and classification, utilizing varied approaches such as radar, computer vision, radio-frequency, and audio. Furthermore, papers concerning UAV payload detection are thoroughly scrutinized, providing a comprehensive understanding of each approach's strengths and weaknesses, all of which demonstrate promising results.

Furthermore, this chapter delves into the audio features conventionally employed for training ML models, covering the time domain, frequency domain, time-frequency domain, and cepstral domain. It elucidates the evolution, characteristics, and applications of different features. Additionally, the chapter investigates various feature extraction tools, available in different formats such as software function libraries, plug-ins for host applications, and standalone software applications. Most of these tools are open-source and can be effectively utilized across modern architectures and platforms.

Chapter 3 commences by presenting an elaborate account of the data collection process (see Section 3.1). This phase is further divided into indoor and outdoor operations, contingent upon the availability of unmanned aerial vehicles (UAVs). Notably, out of the 22 UAVs under consideration, 14 were flown and data was collected outdoors, with the specific locations indicated in Figure 3.1. The remaining eight UAVs were flown and their corresponding data was gathered within the controlled environment of K-SW at Purdue University, West Lafayette, IN. In Section 3.2, comprehensive specifications pertaining to the UAVs, including type, weight, size, and other relevant details, are meticulously delineated.

Subsequently, Section 3.3 provides an extensive overview of the proposed UAV classifier, which capitalizes on audio data for classification purposes. Furthermore, Section 3.4 offers an intricate elucidation of the underlying mechanics of feature extraction, delineating the precise methodology employed for calculating Mel-frequency cepstral coefficients (MFCCs). Finally, Section 3.4 introduces the architectural framework of the proposed convolutional neural

network (CNN) for phase one. Notably, phase two and phase three adopt a similar structure to that of the CNN utilized in the initial two phases. To address concerns of overfitting, an additional dropout layer is incorporated between the two convolutional layers in phase three.

Chapter 4 of this dissertation presents a detailed account of the experimental process, encompassing three distinct phases. Each phase involves a comprehensive dataset evaluation, wherein five distinct Machine Learning models are trained to utilize the collected dataset. Notably, all ML models are trained using default settings and parameters. This step serves the purpose of assessing the dataset's efficiency and establishing a baseline for Deep Learning model training.

The first phase (Section 4.1) entails the collection of audio data from 10 specific classes of UAVs, followed by training a CNN classifier using the acquired dataset. The results of this phase indicate an average training accuracy of 97.43% and an average test loss of 0.085, as derived from 10 separate training sessions.

Moving on to the second phase (Section 4.2) of the experiment, audio data from 15 different classes of UAVs are collected. A CNN classifier is subsequently trained using this expanded dataset, based on the CNN structure developed in the first phase. However, notable modifications include an increased number of filters in both the convolutional and dense layers. The findings reveal an average training accuracy of 98.2% and an average test loss of 0.085, obtained from 10 separate training runs.

The third phase (Section 4.3) focuses on the collection of audio data from 22 distinct classes of UAVs, which is utilized to train another CNN classifier. Notably, the CNN structure in this phase incorporates further enhancements, such as an increased number of filters in the convolutional layers and the inclusion of an additional dropout layer between the two convolutional layers. The outcomes of this phase showcase an average training accuracy of 99.1% and an average test loss of 0.027, based on 10 separate training iterations.

Furthermore, in addition to Accuracy, Precision, Recall, and F1-scores, an array of supplementary evaluation metrics are employed to assess the CNN structure implemented in the third phase. These metrics include the examination of a confusion matrix plot (Figure 4.7) and the utilization of OvR ROC curves (Figure 4.8).

In conclusion, this research study encompasses three distinct experimental phases, which culminated in the collection of audio data from a diverse range of 22 different types of unmanned aerial vehicles (UAVs), spanning from handheld drones to Class I UAVs, as depicted in Figure 4.7. A minimum of 100 data entries were obtained for each UAV class, with each data entry having a duration of 5 seconds. In the absence of an available benchmark dataset, five machine learning (ML) models were trained using the collected dataset to assess its quality. Additionally, these trained ML models served as a foundation for subsequent deep learning (DL) model training. Subsequently, a convolutional neural network (CNN) classifier was trained to utilize the acquired dataset. The resulting 22-class classifier achieved an average accuracy score of 99.1% and an average test loss of 0.027. Furthermore, comprehensive evaluations of the CNN structure in the third phase included the examination of additional performance metrics such as precision, recall, F1-scores, as well as the generation of a confusion matrix plot (Figure 4.7) and one-vs-rest receiver operating characteristic (OvR ROC) curves (Figure 4.8). The collective findings from these evaluation metrics affirm that the proposed classifier exhibits near-flawless performance in accurately classifying UAVs utilizing audio data.

The present study is subject to several limitations, which are outlined as follows:

- The dataset used in this research comprises solely Class I UAVs, thus limiting the generalizability of the findings to other classes of UAVs.
- Data collection was confined to the period between sunrise and sunset due to regulatory requirements and considerations of personal safety. Consequently, variations in UAV audio patterns during other times of the day remain unexplored.
- The collected outdoor data incorporates ambient noises originating from the rural environment, including air traffic, ground traffic, birds, insects, wind, human conversations, and other similar sources. These environmental noises may introduce potential confounding factors in the audio dataset.
- The dataset includes a total of 22 distinct classes, representing 22 different UAVs. While this collection is the largest known audio dataset for UAVs to date, it is important to note

that it does not encompass the entirety of UAVs available from various manufacturers and models.

- The number of data entries for each class ranges from 100 to 138, with each data entry spanning a duration of 5 seconds. The inclusion of additional data for each class would enhance the robustness and reliability of the classifier.

- The proposed classifier is based on a convolutional neural network (CNN) structure, which serves as the underlying foundation for its operation and decision-making process.

These limitations should be taken into account when interpreting the findings of this research and considering its potential implications. Future studies may address these limitations by incorporating diverse classes of UAVs, extending data collection periods, mitigating environmental noise effects, expanding the dataset to include a broader range of UAV models, increasing the volume of data for each class, and exploring alternative classifier architectures beyond CNNs.

The aforementioned limitations highlight potential avenues for future research. As part of these research endeavors, a primary objective is to further enhance the dataset by incorporating additional UAV types and augmenting the volume of data available for each class. Furthermore, an area of interest lies in the implementation of alternative deep learning architectures, such as self-supervised learning and semi-supervised learning, utilizing the comprehensive dataset gathered. By conducting a comparative analysis of the performance exhibited by various model structures, valuable insights can be gleaned, enhancing the understanding and effectiveness of UAV classification methodologies.

REFERENCES

[1] Y. Wang, F. E. Fagian, K. E. Ho, and E. T. Matson, "A feature engineering focused system for acoustic uav detection," in *2021 Fifth IEEE International Conference on Robotic Computing (IRC)*, IEEE, 2021, pp. 125–130.

[2] T. Qiblawi, *A drone attack in abu dhabi could mark a dangerous turning point for the middle east. here's what to know*, https://www.cnn.com/2022/01/18/middleeast/uae-abu-dhabi-houthi-yemen-explainer-intl/index.html, 2022.

[3] M. Laris, *As drone popularity increases, feds look to rein in bad behavior*, https://www.washingtonpost.com/transportation/2021/12/03/drones-flying-prosecutions/, 2021.

[4] B. Taha and A. Shoufan, "Machine learning-based drone detection and classification: State-of-the-art in research," *IEEE Access*, vol. 7, pp. 138 669–138 682, 2019.

[5] 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.

[6] 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.

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

[8] 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.

[9] S. Zhang, C. Zhang, and Q. Yang, "Data preparation for data mining," *Applied artificial intelligence*, vol. 17, no. 5-6, pp. 375–381, 2003.

[10] X. Ying, "An overview of overfitting and its solutions," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1168, 2019, p. 022 022.

[11] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4.

[12] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 161–168.

[13] T. Hastie, R. Tibshirani, and J. Friedman, "Unsupervised learning," in *The elements of statistical learning*, Springer, 2009, pp. 485–585.

[14] M. A. Wiering and M. Van Otterlo, "Reinforcement learning," *Adaptation, learning, and optimization*, vol. 12, no. 3, p. 729, 2012.

[15] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.

[16] 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.

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

[18] 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.

[19] E. Tzeng, J. Hoffman, 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.

[20] 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.

[21] M. Dong, "Convolutional neural network achieves human-level accuracy in music genre classification," *arXiv preprint arXiv:1802.09697*, 2018.

[22] S. R. Park and J. Lee, "A fully convolutional neural network for speech enhancement," *arXiv preprint arXiv:1609.07132*, 2016.

[23] 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.

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

[25] 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.

[26] 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.

[27] 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.

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

[29] 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.

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

[31] 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.

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

[33] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in neural information processing systems*, vol. 27, 2014.

[34] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[35] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss, and D. Eck, "Online and linear-time attention by enforcing monotonic alignments," in *International Conference on Machine Learning*, PMLR, 2017, pp. 2837–2846.

[36] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[37] N.-Q. Pham, T.-S. Nguyen, J. Niehues, M. Müller, S. Stüker, and A. Waibel, "Very deep self-attention networks for end-to-end speech recognition," *arXiv preprint:1904.13377*, 2019.

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

[39] M. Shannon, H. Zen, and W. Byrne, "Autoregressive models for statistical parametric speech synthesis," *IEEE transactions on audio, speech, and language processing*, vol. 21, no. 3, pp. 587–597, 2012.

[40] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[41] R. S. Sutton, A. G. Barto, *et al.*, "Introduction to reinforcement learning," 1998.

[42] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," *arXiv preprint arXiv:1811.12560*, 2018.

[43] L. Kaiser, M. Babaeizadeh, P. Milos, *et al.*, "Model-based reinforcement learning for atari," *arXiv preprint arXiv:1903.00374*, 2019.

[44] S. Whiteson, "Treeqn and atreec: Differentiable tree planning for deep reinforcement learning," 2018.

[45] V. Konda and J. Tsitsiklis, "Actor-critic algorithms," *Advances in neural information processing systems*, vol. 12, 1999.

[46] V. Mnih, A. P. Badia, M. Mirza, *et al.*, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, PMLR, 2016, pp. 1928–1937.

[47] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[48] T. Seno, *Welcome to deep reinforcement learning part 1 : Dqn*, Oct. 2017. [Online]. Available: https://towardsdatascience.com/welcome-to-deep-reinforcement-learning-part-1-dqn-c3cab4d41b6b.

[49] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, 2016.

[50] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

[51] J. AbeSSer, "A review of deep learning based methods for acoustic scene classification," *Applied Sciences*, vol. 10, no. 6, 2020.

[52] 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.

[53] 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.

[54] 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.

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

[56] 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.

[57] J. F. Gemmeke, D. P. Ellis, D. Freedman, *et al.*, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2017, pp. 776–780.

[58] 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.

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

[60] 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.

[61] 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.

[62] 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.

[63] 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.

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

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

[66] 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.

[67] 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.

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

[69] N. Kotecha, "Bach2bach: Generating music using a deep reinforcement learning approach," *arXiv preprint arXiv:1812.01060*, 2018.

[70] 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.

[71] 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.

[72] C. P. Smith, "A phoneme detector," *The Journal of the Acoustical Society of America*, vol. 23, no. 4, pp. 446–451, 1951.

[73] K. Stevens, "Autocorrelation analysis of speech sounds," *The Journal of the Acoustical Society of America*, vol. 22, no. 6, pp. 769–771, 1950.

[74] C. G. Howard, "Speech analysis-synthesis scheme using continuous parameters," *The Journal of the Acoustical Society of America*, vol. 28, no. 6, pp. 1091–1098, 1956.

[75] R. K. Potter and J. C. Steinberg, "Toward the specification of speech," *The Journal of the Acoustical Society of America*, vol. 22, no. 6, pp. 807–820, 1950.

[76] S. Chu, S. Narayanan, and C.-C. J. Kuo, "Environmental sound recognition with time–frequency audio features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 6, pp. 1142–1158, 2009.

[77] Y. Li, X. Zhang, H. Jin, *et al.*, "Using multi-stream hierarchical deep neural network to extract deep audio feature for acoustic event detection," *Multimedia Tools and Applications*, vol. 77, no. 1, pp. 897–916, 2018.

[78] Y. Li, X. Li, Y. Zhang, W. Wang, M. Liu, and X. Feng, "Acoustic scene classification using deep audio feature and blstm network," in *2018 International Conference on Audio, Language and Image Processing (ICALIP)*, IEEE, 2018, pp. 371–374.

[79] N. Takahashi, M. Gygli, and L. Van Gool, "Aenet: Learning deep audio features for video analysis," *IEEE Transactions on Multimedia*, vol. 20, no. 3, pp. 513–524, 2017.

[80] M. H. Rahmani, F. Almasganj, and S. A. Seyyedsalehi, "Audio-visual feature fusion via deep neural networks for automatic speech recognition," *Digital Signal Processing*, vol. 82, pp. 54–63, 2018.

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

[82] 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.

[83] 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.

[84] 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.

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

[86] J. J. Burred, A. Robel, and T. Sikora, "Dynamic spectral envelope modeling for timbre analysis of musical instrument sounds," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 663–674, 2009.

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

[88] 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.

[89] 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.

[90] 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.

[91] 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.

[92] 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.

[93] 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.

[94] 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.

[95] 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.

[96] 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.

[97] 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.

[98] 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.

[99] 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.

[100] 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.

[101] 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.

[102] 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.

[103] 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.

[104] 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.

[105] 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.

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

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

[108] 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.

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

[110] L. Grama and C. Rusu, "Audio signal classification using linear predictive coding and random forests," in *2017 International conference on speech technology and human-computer dialogue (SpeD)*, IEEE, 2017, pp. 1–9.

[111] E. Tsau, S.-H. Kim, and C.-C. J. Kuo, "Environmental sound recognition with celp-based features," in *ISSCS 2011-International Symposium on Signals, Circuits and Systems*, IEEE, 2011, pp. 1–4.

[112] A. Sarkar and T. Sreenivas, "Dynamic programming based segmentation approach to lsf matrix reconstruction," in *Ninth European Conference on Speech Communication and Technology*, 2005.

[113] 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.

[114] 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.

[115] 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.

[116] 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.

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

[118] 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.

[119] 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.

[120] 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.

[121] 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.

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

[123] J. Chen, Y. Huang, Q. Li, and K. K. Paliwal, "Recognition of noisy speech using dynamic spectral subband centroids," *IEEE signal processing letters*, vol. 11, no. 2, pp. 258–261, 2004.

[124] N. C. Maddage, C. Xu, and Y. Wang, "A svm c based classification approach to musical audio," 2003.

[125] A. Bernard and A. Alwan, "Source and channel coding for remote speech recognition over error-prone channels," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 01CH37221)*, IEEE, vol. 4, 2001, pp. 2613–2616.

[126] T. Kinjo and K. Funaki, "On hmm speech recognition based on complex speech analysis," in *IECON 2006-32nd Annual Conference on IEEE Industrial Electronics*, IEEE, 2006, pp. 3477–3480.

[127] N. Dave, "Feature extraction methods lpc, plp and mfcc in speech recognition," *International journal for advance research in engineering and technology*, vol. 1, no. 6, pp. 1–4, 2013.

[128] P. J. Clemins and M. T. Johnson, "Generalized perceptual linear prediction features for animal vocalization analysis," *The Journal of the Acoustical Society of America*, vol. 120, no. 1, pp. 527–534, 2006.

[129] M. Glodek, S. Tschechne, G. Layher, *et al.*, "Multiple classifier systems for the classification of audio-visual emotional states," in *International Conference on Affective Computing and Intelligent Interaction*, Springer, 2011, pp. 359–368.

[130] J. Koehler, N. Morgan, H. Hermansky, H.-G. Hirsch, and G. Tong, "Integrating rasta-plp into speech recognition," in *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, vol. 1, 1994, pp. I–421.

[131] D. Hardt and K. Fellbaum, "Spectral subtraction and rasta-filtering in text-dependent hmm-based speaker verification," in *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, IEEE, vol. 2, 1997, pp. 867–870.

[132] Y.-M. Zeng, Z.-Y. Wu, T. Falk, and W.-Y. Chan, "Robust gmm based gender classification using pitch and rasta-plp parameters of speech," in *2006 International conference on machine learning and cybernetics*, IEEE, 2006, pp. 3376–3379.

[133] D. D. Greenwood, "A cochlear frequency-position function for several species29 years later," *The Journal of the Acoustical Society of America*, vol. 87, no. 6, pp. 2592–2605, 1990.

[134] X. Valero and F. Alias, "Gammatone cepstral coefficients: Biologically inspired features for non-speech audio classification," *IEEE Transactions on Multimedia*, vol. 14, no. 6, pp. 1684–1689, 2012.

[135] H. Yin, V. Hohmann, and C. Nadeu, "Acoustic features for speech recognition based on gammatone filterbank and instantaneous frequency," *Speech communication*, vol. 53, no. 5, pp. 707–715, 2011.

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

[137] M. A. A. Zulkifly and N. Yahya, "Relative spectral-perceptual linear prediction (rasta-plp) speech signals analysis using singular value decomposition (svd)," in *2017 IEEE 3rd International Symposium in Robotics and Manufacturing Automation (ROMA)*, IEEE, 2017, pp. 1–5.

[138] Librosa, *Feature extraction*, https://librosa.org/doc/main/feature.html, Mar. 2022.

[139] Marsyas, *Marsyas*, http://marsyas.info/index.html, Mar. 2022.

[140] C. McKay, I. Fujinaga, and P. Depalle, "Jaudio: A feature extraction library," in *Proceedings of the international conference on music information retrieval*, 2005, pp. 600–3.

[141] P. M. Brossier, "The aubio library at mirex 2006," *Synthesis*, 2006.

[142] D. Bogdanov, N. Wack, E. Gómez Gutiérrez, *et al.*, "Essentia: An audio analysis library for music information retrieval," in *Britto A, Gouyon F, Dixon S, editors. 14th Conference of the International Society for Music Information Retrieval (ISMIR); 2013 Nov 4-8; Curitiba, Brazil.[place unknown]: ISMIR; 2013. p. 493-8.*, International Society for Music Information Retrieval (ISMIR), 2013.

[143] J. Bullock and U. Conservatoire, "Libxtract: A lightweight library for audio feature extraction," in *ICMC*, Citeseer, 2007.

[144] B. Mathieu, S. Essid, T. Fillon, J. Prado, and G. Richard, "Yaafe, an easy to use and efficient audio feature extraction software.," in *ISMIR*, Citeseer, 2010, pp. 441–446.

[145] O. Lartillot and P. Toiviainen, "A matlab toolbox for musical feature extraction from audio," in *International conference on digital audio effects*, Bordeaux, vol. 237, 2007, p. 244.

[146] G. Peeters, B. L. Giordano, P. Susini, N. Misdariis, and S. McAdams, "The timbre toolbox: Extracting audio descriptors from musical signals," *The Journal of the Acoustical Society of America*, vol. 130, no. 5, pp. 2902–2916, 2011.

[147] C. Aker and S. Kalkan, "Using deep networks for drone detection," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, 2017, pp. 1–6.

[148] A. Rozantsev, V. Lepetit, and P. Fua, "Detecting flying objects using a single moving camera," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 5, pp. 879–892, 2016.

[149] D. Lee, W. G. La, and H. Kim, "Drone detection and identification system using artificial intelligence," in *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, IEEE, 2018, pp. 1131–1133.

[150] M. Saqib, S. D. Khan, N. Sharma, and M. Blumenstein, "A study on detecting drones using deep convolutional neural networks," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, 2017, pp. 1–5.

[151] R. Yoshihashi, T. T. Trinh, R. Kawakami, S. You, M. Iida, and T. Naemura, "Differentiating objects by motion: Joint detection and tracking of small flying objects," *arXiv preprint arXiv:1709.04666*, 2017.

[152] J. Peng, C. Zheng, P. Lv, T. Cui, Y. Cheng, and S. Lingyu, "Using images rendered by pbrt to train faster r-cnn for uav detection," 2018.

[153] E. Unlu, E. Zenou, and N. Riviere, "Using shape descriptors for uav detection," *Electronic Imaging*, vol. 2018, no. 9, pp. 128–1, 2018.

[154] G. J. Mendis, T. Randeny, J. Wei, and A. Madanayake, "Deep learning based doppler radar for micro uas detection and classification," in *MILCOM 2016-2016 IEEE Military Communications Conference*, IEEE, 2016, pp. 924–929.

[155] B. K. Kim, H.-S. Kang, and S.-O. Park, "Drone classification using convolutional neural networks with merged doppler images," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 1, pp. 38–42, 2016.

[156] P. Zhang, L. Yang, G. Chen, and G. Li, "Classification of drones based on micro-doppler signatures with dual-band radar sensors," in *2017 Progress in Electromagnetics Research Symposium-Fall (PIERS-FALL)*, IEEE, 2017, pp. 638–643.

[157] L. Fuhrmann, O. Biallawons, J. Klare, R. Panhuber, R. Klenke, and J. Ender, "Micro-doppler analysis and classification of uavs at ka band," in *2017 18th International Radar Symposium (IRS)*, IEEE, 2017, pp. 1–9.

[158] B. Torvik, K. E. Olsen, and H. Griffiths, "Classification of birds and uavs based on radar polarimetry," *IEEE geoscience and remote sensing letters*, vol. 13, no. 9, pp. 1305–1309, 2016.

[159] M. I. Skolnik, *Radar handbook*. McGraw-Hill Education, 2008.

[160] Z. Shi, M. Huang, C. Zhao, L. Huang, X. Du, and Y. Zhao, "Detection of lssuav using hash fingerprint based svdd," in *2017 IEEE International Conference on Communications (ICC)*, IEEE, 2017, pp. 1–5.

[161] P. Nguyen, M. Ravindranatha, A. Nguyen, R. Han, and T. Vu, "Investigating cost-effective rf-based detection of drones," in *Proceedings of the 2nd workshop on micro aerial vehicle networks, systems, and applications for civilian use*, 2016, pp. 17–22.

[162] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Micro-uav detection and classification from rf fingerprints using machine learning techniques," in *2019 IEEE Aerospace Conference*, IEEE, 2019, pp. 1–13.

[163] C. Zhao, M. Shi, Z. Cai, and C. Chen, "Detection of unmanned aerial vehicle signal based on gaussian mixture model," in *2017 12th International Conference on Computer Science and Education (ICCSE)*, IEEE, 2017, pp. 289–293.

[164] I. Güvenç, O. Ozdemir, Y. Yapici, H. Mehrpouyan, and D. Matolak, "Detection, localization, and tracking of unauthorized uas and jammers," in *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, IEEE, 2017, pp. 1–10.

[165] A. Bernardini, F. Mangiatordi, E. Pallotti, and L. Capodiferro, "Drone detection by acoustic signature identification," *Electronic Imaging*, vol. 2017, no. 10, pp. 60–64, 2017.

[166] Y. Seo, B. Jang, and S. Im, "Drone detection using convolutional neural networks with acoustic stft features," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, IEEE, 2018, pp. 1–6.

[167] X. Yue, Y. Liu, J. Wang, H. Song, and H. Cao, "Software defined radio and wireless acoustic networking for amateur drone surveillance," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 90–97, 2018.

[168] B. Yang, E. T. Matson, A. H. Smith, J. E. Dietz, and J. C. Gallagher, "Uav detection system with multiple acoustic nodes using machine learning models," in *2019 Third IEEE International Conference on Robotic Computing (IRC)*, IEEE, 2019, pp. 493–498.

[169] S. Jeon, J.-W. Shin, Y.-J. Lee, W.-H. Kim, Y. Kwon, and H.-Y. Yang, "Empirical study of drone sound detection in real-life environment with deep neural networks," in *2017 25th European Signal Processing Conference (EUSIPCO)*, IEEE, 2017, pp. 1858–1862.

[170] J. Kim, C. Park, J. Ahn, Y. Ko, J. Park, and J. C. Gallagher, "Real-time uav sound detection and analysis system," in *2017 IEEE Sensors Applications Symposium (SAS)*, IEEE, 2017, pp. 1–5.

[171] I. Ku, S. Roh, G. Kim, C. Taylor, Y. Wang, and E. T. Matson, "Uav payload detection using deep learning and data augmentation," in *2022 Sixth IEEE International Conference on Robotic Computing (IRC)*, IEEE, 2022, pp. 18–25.

[172] F. Fioranelli, M. Ritchie, H. Griffiths, and H. Borrion, "Classification of loaded/unloaded micro-drones using multistatic radar," *Electronics Letters*, vol. 51, no. 22, pp. 1813–1815, 2015.

[173] L. Pallotta, C. Clemente, A. Raddi, and G. Giunta, "A feature-based approach for loaded/unloaded drones classification exploiting micro-doppler signatures," in *2020 IEEE Radar Conference (RadarConf20)*, IEEE, 2020, pp. 1–6.

[174] Z. Chen and X. Li, "New field crop phenotying robots phenobee and phenovacbot," 2022. [Online]. Available: https://engineering.purdue.edu/ABEPlantSensorLab/news/archer-chen-and-xuan-li-successfully-demonstrated-their-new-field-crop-phenotying-robots-phenobee-and-phenovacbot-in-purdues-2022-digital-agriculture-showcase-on-sep-8th-2022.

[175] *Autel evo 2*, https://www.autelrobotics.com/productdetail/evo-ii-drones.html, Accessed: 2023-04-11.

[176] *Yuneec typhoon h plus*, https://https://us.yuneec.com/typhoon-h-plus/, Accessed: 2023-04-11.

[177] *Swellpro splash 3 plus*, https://https://us.yuneec.com/typhoon-h-plus/, Accessed: 2023-04-11.

[178] *DJI matrice 200*, https:https://www.dji.com/matrice-200-series, Accessed: 2023-04-11.

[179] *DJI matrice 200 v2*, https://www.dji.com/matrice-200-series-v2, Accessed: 2023-04-11.

[180] *DJI mavic air 2*, https://www.dji.com/mavic-air-2, Accessed: 2023-04-11.

[181] *DJI mavic mini*, https://www.dji.com/mavic-mini, Accessed: 2023-04-11.

[182] *DJI mini 2*, https://www.dji.com/mini-2, Accessed: 2023-04-11.

[183] *DJI mavic 2 pro*, https://www.dji.com/mavic-2, Accessed: 2023-04-11.

[184] *DJI air 2s*, https://www.dji.com/air-2s?gclid=CjwKCAjwitShBhA6EiwAq3RqA8qTU8TBmQHm6anDhDEDvB39fmAdAIuV5Usz6IPUezafcEvleytZFxoCgXIQAvD_BwE, Accessed: 2023-04-11.

[185] *DJI phantom 2*, https://https://www.dji.com/phantom-2, Accessed: 2023-04-11.

[186] *DJI phantom 4*, https://https://www.dji.com/phantom-4, Accessed: 2023-04-11.

[187] *DJI robomaster tt tello talent*, https://https://www.dji.com/robomaster-tt, Accessed: 2023-04-11.

[188] *Hasakee q11*, https://www.amazon.com/HASAKEE-Quadcopter-Skyquad-Protect-Beginners/dp/B0BGH5GSWW, Accessed: 2023-04-11.

[189] *Syma x5sw*, https://www.amazon.com/Cheerwing-X5SW-V3-Explorers2-Headless-Quadcopter/dp/B011JV9HA2?th=1, Accessed: 2023-04-11.

[190] *Syma x20*, https://www.amazon.com/Cheerwing-Control-Quadcopter-Altitude-Take-Off/dp/B06WGVP9FG?th=1, Accessed: 2023-04-11.

[191] *Syma x20p*, https://www.amazon.com/SYMA-Drone-X20P-Landing-Degree/dp/B0B5VKBG7F?source=ps-sl-shoppingads-lpcontext&ref_=fplfs&psc=1&smid=ATVPDKIKX0DER, Accessed: 2023-04-11.

[192] *Syma x26*, https://www.amazon.com/Infrared-Obstacle-Avoidance-Control-Aircraft/dp/B0876PXPQ4, Accessed: 2023-04-11.

[193] *UDI u46*, https://www.amazon.com/UDI-RC-Hovering-Headless-Quadcopter/dp/B0771HCZNH, Accessed: 2023-04-11.

[194] E. L. Salomons and P. J. Havinga, "A survey on the feasibility of sound classification on wireless sensor nodes," *Sensors*, vol. 15, no. 4, pp. 7462–7498, 2015.

[195] S. Hershey, S. Chaudhuri, D. P. Ellis, *et al.*, "Cnn architectures for large-scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, IEEE, 2017, pp. 131–135.

[196] Y. Wang, Z. Chu, I. Ku, E. C. Smith, and E. T. Matson, "A large-scale uav audio dataset and audio-based uav classification using cnn," in *2022 Sixth IEEE International Conference on Robotic Computing (IRC)*, IEEE, 2022, pp. 186–189.