Comparative Analysis of Mel Spectrograms and Short-Time Fourier Transform (STFT) Spectrograms for UAV Audio Detection and Identification Using Deep Learning Models

1st Ravi Gadgil
Department of Computer Science
San Jose State University
San Jose, CA, USA
ravi.gadgil@sjsu.edu

2nd Helen Lei
College of Computing and Information Science
Cornell University
Ithaca, NY, USA
hl883@cornell.edu

3rd Apurva Anand
Department of Computing Sciences
TAMUCC
Corpus Christi, TX, USA
aanand@islander.tamucc.edu

4th Sandeep Amgothu
Department of Computing Sciences
TAMUCC
Corpus Christi, TX, USA
samgothu@islander.tamucc.edu

5th Dulal Kar
Department of Computing Sciences
TAMUCC
Corpus Christi, TX, USA
Dulal.Kar@tamucc.edu

***Abstract*—Unmanned aerial vehicles (UAVs), or drones, offer immense potential but also pose major security concerns due to their accessibility and misuse. Effective detection and identification of drones is extremely important in the mitigation of these risks. This study explores the application of different preprocessing techniques and deep learning models for the identification of drones and detection of unknown drones by their acoustic signature. However, the success of deep learning relies heavily on a large dataset. To address this, this work combines various acoustic files from numerous sources and utilizes data augmentation techniques. We also compared the effectiveness specifically of using a Mel versus STFT based approach in generating audio spectrograms for deep learning multi-class classification. Our findings demonstrate the efficacy of deep learning models in achieving promising results for audio identification of drones. This work focused specifically on CNN, RNN, and CRNN models, and analyzed the potential of each model in combination with Mel and STFT spectrograms. We found that Mel spectrograms yielded the best overall classification results. Our application of entropy thresholding was also very successful in detecting unknown drones, resulting in robust Mel models that were able to classify known and unknown drones fairly well. In conclusion, this analysis compares the immense, varying potential of utilizing different acoustic features and deep learning algorithms for accurate, real-time UAV identification.**

***Index Terms*—drone, UAV, drone audio, drone detection, drone identification, CNN, RNN, CRNN, entropy thresholding, data augmentation, deep learning, machine learning, artificial intelligence, Mel, STFT, spectrogram**

I. Introduction

In recent years, drones have transcended their traditional military applications and have become pervasive across various industries and recreational pursuits. From facilitating last-mile goods delivery to revolutionizing agriculture, surveying, emergency response, and cinematography, drones have rapidly integrated into commercial and commodity sectors [6]. The global drone market is poised for exponential growth, forecasted to surge to 13 billion US dollars by 2025, marking a remarkable 200% expansion within a mere five years [7]. Although some nations have imposed bans on drone sales and usage, many countries facilitate straightforward drone acquisition, subject to compliance with stipulated regulations [8]. However, the widespread availability of drones also brings forth concerns regarding potential misuse, ranging from illicit activities such as drug smuggling to spying on sensitive locations such as military sites or civilian homes. In light of escalating reports documenting such incidents in recent years, there is an urgent need to explore effective methods for drone identification to ascertain if the UAV in a particular airspace is authorized or hostile. Detecting and identifying flying drones poses a significant challenge due to their high speed and diverse shapes, especially when they are small in size and traditional detection methods, such as RF or optical sensors, are used. However, different drone models often exhibit distinctive acoustic signatures which presented an opportunity to enhance identification systems through UAV acoustic recognition technology. Audio signals contain a spectrum of frequencies that can be visualized using Mel spectrograms and Short-Time Fourier Transform (STFT) spectrograms. The features in these signal representations vary according to the sounds generated by different types of UAVs and other audio sources which enables differentiation through pattern recognition. Consequently, we extracted feature sets from these spectrograms and leveraged them to train Deep Learning (DL)

National Science Foundation (NSF)

neural networks in order to identify specific drone models within a Multi-Class Classification (MCC) framework. Then, we compared the performance of models using Mel spectrograms versus STFT spectrograms to establish which image type is more effective for drone identification. Additionally, we used entropy thresholding to develop robust models that could identify known drones in our dataset and detect unknown drones. This approach not only addresses current challenges in drone management, but also lays the groundwork for advancing drone detection capabilities amid evolving technological landscapes.

II. Related Work

Several methods of drone detection have been studied in prior research, including radar, LiDAR, and computer vision. However, these methods suffer from various shortfalls that hinder their efficacy in real-world scenarios. For example, radar excels when detecting large objects such as aircraft and missiles [9]. However, this isn't as useful when attempting to identify smaller and more maneuverable objects like commercial drones that can be easily adapted for nefarious uses. In addition, LiDAR is highly dependent on the weather conditions, so it won't be as effective when faced with turbulent environmental factors [10]. Similarly, computer vision methods are significantly hampered by adverse weather due to limited visibility. Also, they have difficulty in congested urban environments where numerous objects in a small location obscure drone detection [11]. These factors underscore the need for an alternative detection technology that is capable of overcoming these specific challenges in diverse operational environments.

Audio detection is a newer method that holds a lot of potential and can be relatively cheaper then methods like radar [1]. Through analyzing and mining drone sound samples can also overcome the inability of techniques relying on sensors or lighting which involve extensive surveillance to be efficient over large areas [20] as well as are too sensitive to environmental factors [18]. Decoded audio information can also be further used in drone eviction systems [19]. As a result, we focused on drone detection through audio processing.

There are many different features that can be extracted from audio data. The process involves using a Hamming window to reduce edge effects, then extracting various features from each subframe of the windowed signal. These features include Short-Time Energy, Temporal Centroid, Spectral Centroid, Spectralroll-off, and Mel frequency coefficients. By analyzing their acoustic signatures, they are incredibly useful in UAV detection [2]. Linear Predictive Coding is another method for extracting and analyzing audio features that has proven useful in analyzing speech. Upon application to drone detection, the combination of Linear Predictive Coding with the Slope of the Frequency Parameter and the Zero Crossing Rate yielded strong results in detecting larger drones [15].

However drone detection has now further evolved to incorporate Machine Learning techniques. With the goal of optimizing the detection of drones through audio, many previous works have explored the extraction of Mel-frequency cepstral coefficients (MFCC) features as well as Short-Time Fourier transform (STFT) in processing drone audio clips. Study [4] compares a STFT-based CNN approach to GMM and RNN models using MFCCs and Mel spectrograms on a dataset that included various background noises, and found the STFT-based CNN achieved lower false positive rates for signals similar to drones, highlighting the effectiveness of the use of STFT features and a CNN model for audio drone detection in complex environments.

The use of analyzing generated spectrogram images has also proved to hold great potential in the use of Audio Drone Detection and machine learning techniques, such as K Nearest Neighbors classification [17]. Further studies analyze the use of these images in combination with deep learning techniques. A solution to achieve more generalizable results is to account for unknown drones in the classification as well. Using the DronePrint prediction pipeline, the model was able to classify known drones with an accuracy of 95% and unknown drones with an accuracy of 86% [14]. We attempted to raise the unknown drone accruacy through the use of different techniques, such as entropy thresholding. Additionally, we incorporated the Crazyflie drone model into our identification dataset which marks its introduction to this domain and expands the scope of research to include a previously unutilized drone model.

Comparing the effectiveness of both STFT and MFCC approaches as well as traditional machine learning versus deep learning techniques, study [3] worked with both STFT and MFCC features in combination with SVM and CNN classification models. They found that MFCC-SVM was unable to distinguish between plane and drone sounds well and the best configuration uses STFT-SVM. CNN didn't work as well for this task since it only supported binary classification and required many training resources. However, deep learning models still have a lot of potential if applied to expansive datasets and multi-class classification.

With the use of deep learning models, a large dataset is required. To address the scarcity of drone sound data, study [5] explores data augmentation to synthesize realistic training data. This is helpful as many works, such as [3], faced limitations as they struggled with data collection and the results would not be very applicable in more generalized settings. Frameworks like Generative Adversarial Networks (GANs) can be used to augment data [13].

As a result, our work focused on different techniques to expand current datasets with audio data from previously unused drone models. We used these expanded datasets to improve results achieved by previous works utilizing deep learning to classify drones based on Mel and STFT features. Moreover, we conducted a pioneering comparative analysis of these audio-based drone identification approaches which provides a first-time evaluation of their relative effectiveness and performance instead of relying on a single method. Finally, we contributed the use of entropy thresholding to develop robust Deep Learning systems that are capable of both detecting unknown drones and identifying known UAV models.

III. Proposed Methodology

This project consists of four main phases: data collation, preprocessing and feature extraction, implementation and training of deep learning models using the dataset, and the evaluation and comparison of models based on various performance metrics.

A. Data Acquistion

The scarcity of drone audio data poses a significant challenge in UAV acoustic detection tasks employing Deep Learning. Consequently, we made extensive efforts to synthesize a dataset of adequate size to train the models. Here are the four main avenues that we used to collate data:

Experimentally-Captured Drone Sounds: Utilizing the built-in microphone feature of the iPhone 13, we meticulously recorded drone acoustic profiles, covering a spectrum of drone classes including CrazyFlie and DJI Phantom 4 Pro. Flying drones were carefully orchestrated at 1 meter above ground within a 1-meter radius.

YouTube-Sourced Drone Sounds: The dataset was augmented further with drone acoustic profiles sourced from YouTube videos. This segment encompasses a diverse array of drone models, including DJI Mavic Pro and DJI Matrice 100.

Online Databases-Sourced Drone Sounds: Supplementary drone acoustic profiles were curated from reputable online databases, featuring models like Parrot Bebop and Parrot Mambo.

Non-Drone Sounds: To ensure robustness against irrelevant noise in congested environments, non-drone acoustic profiles were gathered from experimental recordings and online databases. These audio clips captured ambient sounds such as airplane noise, bird chirps, calm environments, human speech, rainfall, and traffic.

Pitch Shifting: To address imbalances in data amongst the drone classes, the pitch shifting tool in Python's librosa library was used to augment the samples of the following minority classes: Crazyflie, DJI MavicPro, DJI Matrice 100, and DJI Phantom 4 Pro. The semitones used to generate new audio clips for each class are described in the next section.

This meticulous dataset compilation process ensures the model's robustness and ability to generalize across various drone models and acoustic environments.

B. Audio Dataset Description

After using the four methods of data collection described above, we created a dataset of labeled one-second audio clips that were ready for preprocessing. The training corpus had a total of 3088 samples with 1654 drone samples and 1434 non-drone samples. The test dataset had a total of 768 samples with 411 drone samples and 357 non-drone samples. Consequently, we used an 80%-20% train-test split for our dataset. Here is a statistical breakdown of this dataset for each class:

Audio Class	# Train Samples	# Test Samples
Bebop	420	104
Crazyflie	210	52
Mambo	420	104
Matrice100	200	50
Mavic	200	50
NonDrone	1434	357
Phantom	204	51

Table I: The number of audio training and testing samples in each class.

The Bebop and Mambo classes have a higher number of training and testing samples relative to other classes because they contain both pure and mixed files. The pure clips feature only drone noise, whereas the mixed samples contain drone audio mixed with background non-drone noises. This is done to increase the robustness of the model for practical scenarios in crowded environments.

The number of training and testing samples for Crazyflie was increased by applying pitch shifts of -1 and +1 semitones. Additionally, the minority drone classes Matrice100, Mavic, and Phantom were augmented with pitch shifts of -2, -1, +1, and +2 semitones.

C. Data Preprocessing

- **Short-Time Fourier transform (STFT)** is one method of feature extraction in processing audio data. The audio is segmented and then each segment is processed separately to compute the STFT and extract the magnitude. The formula used is given as follows:

$$X(m,k) = \sum_{n=0}^{N-1} x(n+mH) \cdot w(n) \cdot e^{-i\frac{2\pi}{N}nk}$$

  where:
  - $X(m,k)$ is the STFT of the signal.
  - $x(n+mH)$ is the input signal at frame m and sample n, with a hop size H.
  - $w(n)$ is the window function.
  - $e^{-i\frac{2\pi}{N}nk}$ transforms the signal to the frequency domain.
  - N is the FFT window size.

  We used the Librosa library to generate STFT Spectrograms. The window function used is the Hann function. The main parameters we focused on were the sampling rate, window size, and the hop size. We experimented with general sampling rates and found 16000 Hz to yield the best results. We also found that a larger frame size would result in higher frequency resolution but lower time resolution while a smaller frame size would result in lower frequency resolution but higher time resolution, and a smaller hop length would provide even higher time resolution. As a result, we found that the most optimal sample size was 2048 for the window length and 256 for the hop length. This maintained a relatively high frequency resolution while balancing out the lower time resolution with a higher time resolution resulting from lower hop length.

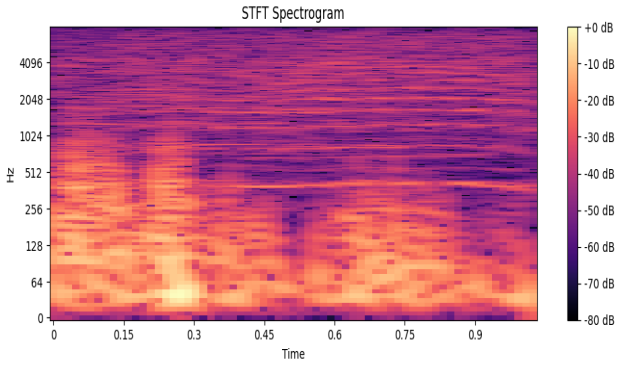

Fig. 1: An example of a STFT spectrogram representation of drone noise taken from a audio clip of a Bebop drone.

- **Mel spectrograms** contain valuable features such as the frequency and power of a signal over time. These features can be used to build a robust deep learning model. Meticulous data preprocessing techniques are employed to compute these spectrograms which were used to train the models. The formula to compute these features is provided here:

$$Mel_{spec}[m,k] = \sum_{n=0}^{N/2} \log(H_i(k) \cdot |X(m,k)|^2)$$

where:

  - $Mel_{spec}[m,k]$ is the Mel spectrogram of the audio signal.
  - $H_i(k)$ is the i-th Mel filterbank at discrete frequency index bin k.
  - $X(m,k)$ is the STFT of the signal.
  - N is the FFT window size.

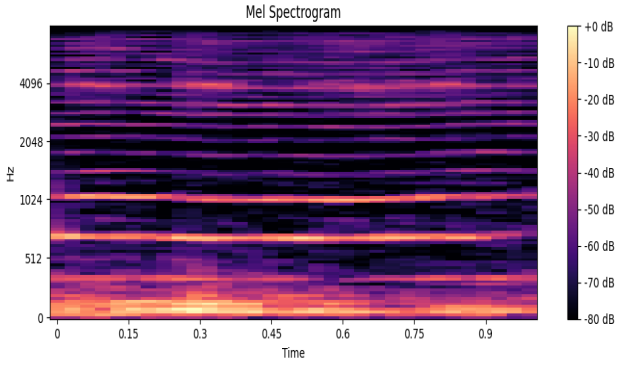

Fig. 2: An example of drone noise in Mel spectrogram representation. The drone model is Crazyflie.

Mel Spectrogram Extraction: Mel spectrograms are renowned for their effectiveness in capturing essential characteristics of audio signals. They are extracted from the raw audio files using the librosa library. This process facilitates spectrogram creation with optimal efficiency and accuracy. Key parameters governing the Mel spectrogram extraction process include:

- **Sample Rate:** The audio sample rate, set at 16000 Hz, ensures adequate representation of the audio signals.
- **Duration:** Audio clips, each spanning 1 second, provide ample temporal context for feature extraction.
- **Number of Mel Bands:** A total of 128 Mel bands is computed to encapsulate ranges of frequencies that are perceptually and spectrally relevant.
- **FFT Window Size and Hop Length:** The FFT window size, set at 2048, and the hop length of 256 frames are pivotal in balancing frequency and time resolution during the spectrogram computation.

D. Model Architecture and Evaluation Metrics

We evaluated three different model architectures for both image datasets: Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Convolutional Recurrent Neural Networks (CRNN).

CNN: The specific CNN architecture is a sequential model comprised of the following layers:

- **Convolutional layers**: These layers perform convolutions, which are operations that identify and extract patterns within the audio data. The model utilizes three convolutional layers that have 32, 64, and 128 filters, each using a 3x3 kernel. These filters operate use the ReLU (Rectified Linear Unit) activation function.
- **Max pooling layers**: Following each convolutional layer is a max pooling layer with a pool size of 2×2. This layer downsamples the data, reducing its dimensionality while retaining the most significant features.
- **Flatten layer**: This layer transforms the extracted features from a 2D format into a 1D vector suitable for feeding into the fully-connected layers.
- **Dense layers**: The model utilizes two fully-connected layers. The first layer has 128 neurons with a ReLU activation function, followed by a dropout layer with a rate of 0.5 to prevent overfitting. The final dense layer has seven neurons with a softmax activation function, corresponding to the seven output class labels.

RNN:

- **Bidirectional recurrent layers**: These layers capture sequential and long-term dependencies in the audio files by processing them in the forward and backward directions. This feature helps enhance contextual understanding. The model utilizes two bidirectional recurrent layers which both have 32 units. These filters use the ReLU activation function.
- **Batch normalization layers**: These layers are inserted after the recurrent layers to address internal covariate shift.
- **Dense layers**: The model utilizes two fully-connected layers. The first layer has 32 neurons with a ReLU activation function, followed by a dropout layer with a rate of 0.3 to prevent overfitting. The final dense

layer has 7 neurons with a softmax activation function, corresponding to the seven output class labels.

CRNN: This model combines the CNN model with one bidirectional recurrent layer on the top to synthesize the advantages of convolutions and long-term dependencies for enhanced performance. The architecture is constructed by first removing the Dense layers from the CNN. Next, a bidirectional recurrent layer with 32 units and ReLU activation is added, followed by a batch normalization layer. Finally, the architecture is completed using the dense layers from the RNN model described above.

The performance of the models are evaluated using the following metrics: accuracy, F1-score, precision, and recall.

E. Experimental Framework

We trained and evaluated three model architectures on distinct datasets of images generated using two preprocessing techniques that were applied to the corpus of audio files. The first dataset consists of spectrograms generated through Short-Time Fourier Transform preprocessing that is employed on each audio sample in the dataset. The second dataset consists of spectrograms generated using Mel features. The number of spectrograms in both datasets is equal to the number of files in the audio dataset as one spectrogram is created for each sample.

We then tested the Mel and STFT spectrograms generated in combination with the CNN, RNN, and CRNN models outlined in Model Architecture.

Name	Data Type	Total Parameters
CNN-STFT	STFT Specs	3,723,479
RNN-STFT	STFT Specs	402,711
CRNN-STFT	STFT Specs	7,343,831
CNN-Mel	Mel Specs	1,241,159
RNN-Mel	Mel Specs	134,407
CRNN-Mel	Mel Specs	2,398,023

Table II: The names of the models followed by their training image format and total number of parameters.

For each experiment, we kept the model parameters uniform and only changed parameters in the dataset for accurate comparison. The height and width of the two spectrogram image types were 128 and 55.5 respectively. The width of the images is calculated using the following formula:

$$w = 1 + \left\lfloor \frac{(T \cdot r) - N}{H} \right\rfloor$$

where:

- w is the width of the spectrogram.
- T is the duration of each audio clip in seconds.
- r is the sample rate.
- N is the FFT window size.
- H is the hop length.

The batch size was 32 samples. Each model is compiled using the Adam optimizer to compute adaptive learning rates for the parameters. The loss of each model is evaluated using the categorical crossentropy loss function.Each model was trained for a total of 185 epochs.

F. Unknown Drone Detection

Our Deep Learning (DL) models are capable of identifying six specific drone models based on their acoustic signatures. However, we recognized that these models are extremely vulnerable to unknown drone models. This is a major problem because undisclosed hostile drones could be classified as a friendly model which would compromise the efficacy of our system. Consequently, we implemented a procedure that categorizes the acoustic signatures of foreign drones as "Unknown Drone."

To implement this feature, we started by collating a dataset with unknown drone samples. We extracted 50 one-second audio samples of the drone model DJI Spark from YouTube videos. Then, we applied pitch shifts of -2, -1, +1, and +2 semitones to expand the dataset to a total of 250 samples. Lastly, these audio samples were used to generate Mel and STFT spectrograms. This dataset was not utilized during the training of our models.

Afterward, we employed a technique called entropy thresholding to detect undisclosed drones. Entropy thresholding is employed in classification tasks to determine the classification of a particular sample based on the model's uncertainty. This procedure is particularly advantageous when working with mixed datasets where classes are not perfectly separable and occasionally overlap [12]. The process begins with calculating the entropy of a sample. This value quantifies the uncertainty of the sample's classification within its corresponding region. Subsequently, this value is compared against a predefined threshold. If the entropy value is less than or equal to the threshold, the sample is classified as belonging to the majority class. Conversely, if the entropy value exceeds the threshold, indicating substantial uncertainty, further processing or alternative labeling may be necessary. The formula used to compute the entropy for a sample is expressed as follows:

$$H(P) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

where:

- $H(P)$ is the entropy of the spectrogram.
- p_i is the i-th class in the probability distribution P.

For our research, samples surpassing the threshold were classified as "Unknown Drone." We determined optimal thresholds for the models by identifying values that achieved 80% accuracy on the unknown drone dataset. Additionally, we tested our models' robustness after implementing this functionality to check if the accuracy of the known classes dropped significantly due to false detections of known UAVs as unknown drones.

IV. Results

A. Overall Experimental Metrics

Model	Accuracy	Precision	Recall	F1-Score
CNN-STFT	96.48%	96.98%	96.35%	97.02%
RNN-STFT	93.62%	93.86%	93.62%	93.80%
CRNN-STFT	95.05%	95.17%	94.92%	94.88%
CNN-Mel	96.74%	96.74%	96.74%	96.95%
RNN-Mel	90.23%	90.67%	88.54%	92.48%
CRNN-Mel	96.48%	96.73%	96.35%	97.23%

Table III: Performance metrics of all models on the test dataset.

The statistics in Table III demonstrated that the CNN was the strongest Mel spectrogram model for this dataset. It is followed by the CRNN and RNN in terms of performance. This is because it outperformed the CRNN-Mel model by 0.26% for accuracy, 0.01% for precision, and 0.39% for recall. Additionally, it significantly surpassed the RNN-Mel model by 6.51% in accuracy, 6.07% in precision, 8.2% in recall, and 4.47% in F1-Score. This pattern continued when observing the STFT models. The CNN was the top-performing STFT-spectrogram model, followed by the CRNN and RNN. It surpassed the CRNN-STFT model by 1.43% for accuracy, 1.81% for precision, 1.43% for recall, and 2.14% for F1-Score. In addition, it outperformed the RNN-STFT model by 2.86% in accuracy, 3.12% in precision, 2.73% in recall, and 3.22% in F1-Score. This trend is positive because the Mel and STFT CNNs utilized 1,156,864 and 3,620,352 fewer parameters respectively relative to the Mel and STFT CRNNs as per Table II. This translated to a reduction of approximately 57.76% and 49.3% in the number of parameters. Furthermore, both CNN models required a significantly reduced training time compared to their CRNN counterparts. Thus, these findings indicate that the CNN offers the highest performance on this dataset with greater computational and resource efficiency.

The CNN and CRNN architectures achieved higher performance due to the format of the dataset as convolutional layers are adept at processing and extracting relevant features from images. However, this isn't the case for RNNs which utilized recurrent layers that are more proficient at processing sequential and time-series data such as text and weather modeling. Also, the audio clips used in the experiment have a short duration of just 1 second which limits the RNN's ability is unable to identify time-dependent patterns or trends. Lastly, the RNN's relatively simpler structure and lack of additional recurrent layers hindered its performance on the dataset.

In comparing the best Mel and STFT models, the Mel systems outperformed the STFT models for this dataset. Firstly, the Mel models achieved the highest scores for accuracy, recall, and F1-Score with the CNN-Mel model scoring 96.74% in accuracy and recall, and the CRNN-Mel model scoring 97.23% in F1-Score. This is also clear when analyzing the results of the strongest model architecture for both datasets: the CNN. The CNN-Mel model surpassed the CNN-STFT model by 0.26% for accuracy and 0.39% for recall. This disparity is even more pronounced when comparing the CRNN models where the CRNN-Mel model exceeded the CRNN-STFT model by 1.43% for accuracy, 1.56% for precision, 1.43% for recall, and 2.35% for F1-Score. In fact, the CRNN-Mel model matched the CNN-STFT model's accuracy and recall while surpassing it in F1-Score. Furthermore, the Mel models also demonstrated greater computational and resource efficiency. CNN-Mel and CRNN-Mel utilized 2,482,320 and 4,945,808 fewer parameters relative to their STFT equivalents. This represents a reduction of approximately 66.67% and 67.35% in the number of parameters with increased performance. Furthermore, the Mel models required less training time compared to the STFT models.

The Mel spectrogram models outperformed the STFT models in both effectiveness and efficiency because Mel spectrograms excel at highlighting perceptually important features while minimizing irrelevant noise. They also streamlined the training process by lowering the dimensionality of audio signals which results in more manageable feature matrices that reduce computational overhead. Conversely, STFT spectrograms are more suited for scenarios requiring detailed linear time-frequency representation. However, this level of detail may be unnecessary for modeling one-second audio clips. Additionally, STFT spectrograms can suffer from spectral leakage where sharp transitions in frequency are blurred out due to the windowing effect. This potentially resulted in the loss of perceptually relevant features and negatively impacted model performance.

B. Class-by-Class Analysis

Class Name	Accuracy	Precision	Recall	F1-Score
Bebop	99.04%	94%	99%	96%
Crazyflie	100%	100%	100%	100%
Mambo	84.62%	91%	85%	88%
Matrice100	100%	100%	100%	100%
Mavic	100%	100%	100%	99%
NonDrone	97.48%	97%	97%	97%
Phantom	98.04%	100%	98%	99%

Table IV: Performance metrics of the CNN-STFT model for each class.

Class Name	Accuracy	Precision	Recall	F1-Score
Bebop	95.19%	91%	95%	93%
Crazyflie	100%	100%	100%	100%
Mambo	75%	88%	75%	81%
Matrice100	94%	98%	94%	96%
Mavic	100%	100%	100%	95%
NonDrone	95.80%	95.8%	96%	95%
Phantom	100%	100%	100%	96%

Table V: Performance metrics of the RNN-STFT model for each class.

Class Name	Accuracy	Precision	Recall	F1-Score
Bebop	97.12%	92%	97%	94%
Crazyflie	100%	98%	100%	99%
Mambo	87.5%	88%	88%	88%
Matrice100	90%	100%	90%	95%
Mavic	96%	100%	96%	98%
NonDrone	96.08%	97%	96%	97%
Phantom	98.04%	89%	98%	93%

Table VI: Performance metrics of the CRNN-STFT model for each class.

When analyzing the class-by-class accuracies for the STFT models alone, it is observed that the CNN-STFT model in Table IV achieved the highest scores for the Bebop, Matrice100, and NonDrone classes by 1.92%, 6%, and 1.4% relative to the second-best models for these categories. Additionally, the RNN-STFT model in Table V uniquely attained a 100% accuracy for the Phantom class, surpassing all other models by 1.96%. All STFT models delivered peak performance for the Crazyflie class. For the Mambo class, the CRNN-STFT Table VI was the best model as it exceeded the next closest model by 2.88%. Lastly, the CNN-STFT and RNN-STFT both earned the highest possible accuracy for the Mavic class, surpassing the CRNN-STFT by 4%.

These results indicate that the CNN-STFT was the best STFT model for drone identification. This is because it achieved the most optimal performance for 5 categories. In addition, it is the second-best model for the Mambo and Phantom classes.

Class Name	Accuracy	Precision	Recall	F1-Score
Bebop	99.04%	94%	99%	96%
Crazyflie	100%	98%	100%	99%
Mambo	87.5%	97%	88%	92%
Matrice100	100%	96%	100%	98%
Mavic	96%	100%	96%	98%
NonDrone	97.76%	97%	98%	97%
Phantom	98.04%	98%	98%	98%

Table VII: Performance metrics of the CNN-Mel model for each class.

Class Name	Accuracy	Precision	Recall	F1-Score
Bebop	97.12%	80%	97%	87%
Crazyflie	100%	100%	100%	100%
Mambo	78.85%	71%	79%	75%
Matrice100	100%	94%	100%	97%
Mavic	100%	96%	100%	98%
NonDrone	86.27%	97%	86%	91%
Phantom	98.04%	100%	98%	99%

Table VIII: Performance metrics of the RNN-Mel model for each class.

Class Name	Accuracy	Precision	Recall	F1-Score
Bebop	96.15%	94%	96%	95%
Crazyflie	100%	100%	100%	100%
Mambo	91.35%	90%	91%	90%
Matrice100	100%	100%	100%	100%
Mavic	100%	98%	100%	99%
NonDrone	96.36%	97%	96%	97%
Phantom	98.04%	100%	98%	99%

Table IX: Performance metrics of the CRNN-Mel model for each class.

We also compared the models within the Mel spectrogram set to identify the best model in this group. For the Bebop and NonDrone classes, the CNN-Mel in Table VII achieved the highest scores and outperformed the second-best Mel models for these categories by 1.92% and 1.4% respectively. It is also worth noting that CNN-Mel's NonDrone accuracy was the highest of all the models in this study. In addition, the CRNN-Mel in Table IX exclusively attained a high score of 91.35% for the Mambo class, surpassing all other models in the study. For the Crazyflie and Matrice100 classes, all Mel models delivered the most optimal performance possible. Conversely, these models achieved the same accuracy for the Phantom class and were unable to reach the peak performance of 100% set by RNN-STFT. Lastly, the RNN-Mel and CRNN-Mel earned the apex score for the Mavic class, surmounting the CNN-Mel by 4%.

These statistical trends demonstrated that the CRNN-Mel was the most effective Mel model for drone identification. This is because it achieved the best scores for 5 drone classes within the Mel model set. In addition, it is exceeded by the second-best CNN-Mel for the Bebop and NonDrone classes by very narrow margins under 2%. In contrast, the CRNN-Mel significantly outperformed the CNN-Mel for two drone classes Mambo and Matrice100 with substantial performance differences of 4% or higher.

Examining broader trends across all models, it is noteworthy that our models consistently classified 6 out of 7 known classes with an accuracy of 90% or higher.Another significant finding was the exceptional performance of the experimentally recorded drone classes Crazyflie and Phantom, which consistently exceeded 98% for all models. In contrast, the Mambo class had the lowest overall accuracy, with only the CRNN-Mel achieving above 90%. This reduced performance was likely caused by the inclusion of both pure and mixed samples for the Mambo class, which produce prominent decibel bands at many different frequencies. As a result, this diversity makes it more challenging for the models to extract patterns and trends from these feature sets.

After comparing the performance of the Mel and STFT models on a class-by-class basis, it was observed that Mel models generally provided a stronger approach for drone identification. This advantage was not immediately clear from a direct comparison of the top models for each dataset. For instance, CNN-STFT outperforms CRNN-Mel for the Bebop and NonDrone classes by 2.89% and 1.12% respectively. Conversely, CRNN-Mel exceeds CNN-STFT for the Mambo

category by 6.73%. The models have the same scores for the other classes. Although CNN-STFT outperforms CRNN-Mel on two classes instead of one, the margin for the NonDrone class isn't very significant at around 1%. However, CRNN-Mel's shortfall for Bebop is significantly better compared to CNN-STFT's deficit for Mambo which exceeds 5%. Furthermore, CRNN-STFT uses 1,325,456 fewer parameters despite having a combined architecture with more layers than the CNN-STFT. This 35.6% reduction in the number of parameters and significantly lower training runtime suggest that the CRNN-Mel is more computationally and resource-efficient. Therefore, CRNN-Mel is likely better suited for practical scenarios requiring rapid drone identification, although the minor performance differences for the NonDrone class should be considered within the broader context of each model's overall capabilities.

Building on this, the performance disparity between the model types becomes even clearer when examining the second-best models for each dataset: CNN-Mel and CRNN-STFT. CNN-Mel surpasses CRNN-STFT for the Bebop, Matrice100, and NonDrone classes by 1.92%, 10%, and 1.68% respectively. The two models have equivalent performance for the other classes. This shows that CNN-Mel was able to notably outperform CRNN-STFT in these specific cases despite not utilizing bidirectional recurrence and 83.09% fewer parameters. Thus, this rigorous analysis of the class-by-class accuracies indicates that Mel models generally show better performance in drone identification compared to STFT models.

C. Unknown Drone Detection Evaluation

Model	Entropy Threshold	Prior Acc	New Acc	Change
CNN-STFT	0.0000006	96.48%	61.2%	-35.28%
CRNN-STFT	0.001111	95.05%	69.01%	-26.04%
CNN-Mel	0.05	96.74%	93.29%	-3.45%
CRNN-Mel	0.12155	96.48%	94.53%	-1.95%

Table X: Optimal entropy threshold value, prior overall accuracy, new overall accuracy, and change in performance for the CNN and CRNN models that detect 80% of unknown drones.

Class Name	Accuracy	Change in Accuracy
Spark (Unknown Drone)	80%	N/A
Bebop	55.8%	-43.24%
Crazyflie	100%	0%
Mambo	51.9%	-32.72%
Matrice100	10%	-90%
Mavic	60%	-40%
NonDrone	65.27%	-32.21%
Phantom	74.51%	-23.53%

Table XI: Class-by-class accuracy of the CNN-STFT model after using entropy thresholding. The second column measures the variation in accuracy for the known classes.

Class Name	Accuracy	Change in Accuracy
Spark (Unknown Drone)	80%	N/A
Bebop	80.77%	-16.35%
Crazyflie	96.15%	-3.85%
Mambo	51.92%	-35.58%
Matrice100	40%	-50%
Mavic	22%	-74%
NonDrone	76.19%	-19.89%
Phantom	76.47%	-21.57%

Table XII: Class-by-class accuracy of the CRNN-STFT model after using entropy thresholding. The second column measures the variation in accuracy for the known classes.

Class Name	Accuracy	Change in Accuracy
Spark (Unknown Drone)	80%	N/A
Bebop	97.12%	-1.92%
Crazyflie	96.15%	-3.85%
Mambo	79.81%	-7.69%
Matrice100	100%	0%
Mavic	92%	-4%
NonDrone	93%	-4.76%
Phantom	92.16%	-5.88%

Table XIII: Class-by-class accuracy of the CNN-Mel model after using entropy thresholding. The second column measures the variation in accuracy for the known classes.

Class Name	Accuracy	Change in Accuracy
Spark (Unknown Drone)	80%	N/A
Bebop	99.04%	+2.89%
Crazyflie	100%	0%
Mambo	88.46%	-2.89%
Matrice100	98%	-2%
Mavic	94%	-6%
NonDrone	93.84%	-2.52%
Phantom	94.12%	-3.92%

Table XIV: Class-by-class accuracy of the CRNN-Mel model after using entropy thresholding. The second column measures the variation in accuracy for the known classes.

When analyzing the broader trends across both model groups, it is evident that Crazyflie demonstrated the highest stability. This is because both groups experienced average losses of just 1.85% when classifying these samples after entropy thresholding. In contrast, the most volatile category for the STFT models was Matrice100 as they experienced an average loss of 70%. For the Mel models, the class with the least robustness was the Mambo class where both models suffered an average loss of 5.29%.

Upon reviewing the revised scores and accuracy changes for the STFT pair alone, it is clear that the STFT models in Tables XI and XII suffered notable performance declines that for all known classes except Crazyflie. These accuracy reductions ranged anywhere from 15% to 90% for 6 out of the 7 known classes. Moreover, both models performed at 60% or below for three classes which include Mambo, Matrice100, and Mavic. This haphazard classification rate represents a potential security threat because the model may be randomly guessing instead of applying learned patterns from the known class dataset to identify samples. In addition, the accuracy losses for both STFT models in Table X both

surpassed 25%, resulting in overall accuracies that are below 70%. Therefore, these statistical observations reveal that the STFT may face major challenges in real-world applications, as the potential for frequent false detections of unknown drones could lead to increased costs from misused defensive measures and interference on the operations of friendly UAVs.

This lack of robustness is attributed to the entropy or uncertainty of STFT sample predictions exceeding the threshold value. As a result, many known samples are incorrectly classified with the label "Unknown Drone." When comparing the STFT models to the Mel systems, the Mel models consistently outperformed the STFT models in 6 out of 7 known classes, due to their greater robustness. This increased stability allowed the Mel models to maintain overall accuracies above 90% and incur losses that were below 4%, as shown in Table X. Moreover, the class-by-class accuracies for the Mel models in Tables XIII and XIV showed that 6 out of 7 categories remained at 90% or higher with the exception of Mambo. This increased performance of the Mel models relative to the STFT systems is a positive finding because the Mel models have a higher resource and computational efficiency due to lower total parameters and training runtime. As a result, the Mel systems can be deployed faster and are capable of detecting both known and unknown drones accurately and more rapidly which is crucial for implementing effective security measures in various contexts.

Moving onto the comparison between the Mel models, CNN-Mel exhibited smaller accuracy losses for the Matrice100 and Mavic classes. However, CRNN-Mel was more robust for other categories in this dataset. Furthermore, CRNN-Mel outperformed CNN-Mel on all known classes except Matrice100 after utilizing entropy thresholding. For some categories, the differences in performance were significant such as a 3.85% gap for Crazyflie and a 8.65% margin for Mambo. Conversely, the differences were below 2% for all other classes. Consequently, CRNN-Mel may be better suited for practical scenarios where Crazyflie or Mambo are friendly or commonly used drone models as reducing false unknown drone detections is crucial to avoid potential costs and wasted countermeasures. However, a CNN-Mel model might be preferred in other contexts because it has 48.24% fewer total parameters and a considerably shorter training time. Thus, the 2% performance margins in other classes may be a reasonable trade-off for faster UAV identification and reduced computational resources.

D. Comparison with Existing Literature

In this section, we compare our models with benchmark models from previous audio-based drone identification studies. Building on the experiment done by Sara El-Amadi [13], we were able to achieve higher accuracies in audio-based drone identification for all three models. Our best CNN model, CNN-Mel, outperformed the drone identification CNN in [13] by 3.8%. Our most effective RNN model, RNN-STFT, exceeded the RNN in [13] by 36.46%. Finally our best CRNN model, CRNN-Mel, outperformed the CRNN in [13] by 4.26%. Additionally, El-Amadi's drone identification experiment used only 3 classes: Bebop, Mambo, and NonDrone. We expanded our work to include 7 classes: Bebop, Mambo, NonDrone, Matrice100, Phantom, Mavic, and Crazyflie.

In addition, all of our models outperformed the LSTM model used by Harini Kolamunna in [14] for the Phantom class by approximately 30%. Additionally, some of our models maintain the LSTM's performance of 100% for the Matrice100 and Mavic classes.

V. Conclusion

In this paper, we compared two audio-based approaches for UAV identification: Mel spectrograms and STFT spectrograms. To address the challenge of limited drone audio data, we constructed a comprehensive dataset that includes recordings from 6 different drone models. These audio files were obtained through experimental recordings, YouTube, online databases, and pitch shifting techniques. This dataset also introduced the smaller and more maneuverable drone model Crazyflie for drone identification tasks. Next, we trained three model architectures on Mel and STFT spectrogram datasets based on this audio dataset. The models demonstrated strong performance by achieving class accuracies of 90% or higher in 6 out of 7 categories. Consequently, our models either maintained or exceeded results reported in existing liteature which shows both generalizability and effectiveness across categories.

Additionally, we introduced entropy thresholding for detecting unknown drones. This method has not been previously applied specifically for drone detection. This approach led to the development of two Mel models that classified unknown drones with an accuracy of 80% while exhibiting high robustness for known classes. As a result, these models maintained scores of 90% or higher for most known categories.

Lastly, we contributed a comparative analysis of two audio-based drone identification approaches which establishes that the Mel spectrogram method is more effective for both drone identification and unknown drone detection. This was first proven through a broader analysis of overall experimental metrics which showed that the Mel models generally demonstrated higher performance due to better feature representation and increased robustness. Furthermore, a rigorous class-by-class comparison emphasized this further as the Mel models mostly maintained or exceeded the categorical accuracies of the STFT models. Moreover, any slight shortfalls of the Mel models for some categories are an acceptable trade-off since they require a considerably lower amount of total parameters and training runtime which boosts efficiency.

For future work, further experimentation can be conducted to boost the unknown drone detection of the Mel models to 90% while having sustained high robustness for known classes. Additionally, feature engineering techniques could be explored to better differentiate features in both pure and mixed samples of the Mambo class for improved pattern recognition. Lastly, adding more drone models to the dataset and utilizing pretrained CNN-based architectures such as the VGG-19 may

contribute to increased performance across a broader range of drone samples.

VI. Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No. 2150351. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] M. A. Khan, H. Menouar, A. Eldeeb, A. Abu-Dayya and F. D. Salim, "On the Detection of Unauthorized Drones—Techniques and Future Perspectives: A Review," in IEEE Sensors Journal, vol. 22, no. 12, pp. 11439-11455, 15 June 15, 2022.

[2] A. Bernardini, F. Mangiatordi, E. Pallotti, and L. Capodiferro, "Drone detection by acoustic signature identification," in *Proceedings of the International Conference on UAV Detection and Identification*, Rome, Italy, pp. 45-50, 2022.

[3] B. Yang, E. T. Manson, A. Smith, E. Diatz, and J. Gallagher, "UAV detection system with multiple acoustic nodes using machine learning models," International Journal of UAV Research, vol. 15, no. 3, pp. 123-135, 2022.

[4] Seo, Yoojeong, Beomhui Jang, and Sungbin Im. "Drone detection using convolutional neural networks with acoustic STFT features." 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE, 2018.

[5] Jeon, Sungho, et al. "Empirical study of drone sound detection in real-life environment with deep neural networks." 2017 25th European Signal Processing Conference (EUSIPCO). IEEE, 2017.

[6] Altawy, Riham, and Amr M. Youssef. "Security, privacy, and safety aspects of civilian drones: A survey." ACM Transactions on Cyber-Physical Systems 1.2 (2016): 1-25.

[7] I. Wagner. [n.d.]. Commercial UAVs - Statistics & Facts. arXiv:https://www.statista.com/topics/3601/commercial-uavs/

[8] Jones, Therese. International commercial drone regulation and drone delivery services. No. RR-1718/3-RC. Santa Monica, CA, USA: RAND, 2017.

[9] M. Benyamin and G. H. Goldman, "Acoustic detection and tracking of a class i UAS with a small tetrahedral microphone array."

[10] A. R. Wagoner, D. K. Schrader, and E. T. Matson, "Towards a vision-based targeting system for counter unmanned aerial systems (CUAS)," in 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), pp. 237–242.

[11] 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," 2019 Third IEEE International Conference on Robotic Computing (IRC), Naples, Italy, 2019, pp. 493-498, doi:10.1109/IRC.2019.00103.

[12] Yazid, H., Basah, S.N., Rahim, S.A. et al. Performance analysis of entropy thresholding for successful image segmentation. Multimed Tools Appl 81, 6433–6450 (2022). https://doi.org/10.1007/s11042-021-11813-z

[13] Al-Emadi, S.; Al-Ali, A.; Al-Ali, A. Audio-Based Drone Detection and Identification Using Deep Learning Techniques with Dataset Enhancement through Generative Adversarial Networks. Sensors 2021, 21, 4953. https://doi.org/10.3390/s21154953

[14] Kolamunna, Harini & Dahanayaka, Thilini & Li, Junye & Seneviratne, Suranga & Thilakarathna, Kanchana & Zomaya, Albert & Seneviratne, Aruna. (2021). DronePrint: Acoustic Signatures for Open-set Drone Detection and Identification with Online Data. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies. 5. 1-31. 10.1145/3448115.

[15] L. Hauzenberger and E. Holmberg Ohlsson, "Drone detection using audio analysis," 2015. [Online]. Available: https://lup.lub.lu.se/student-papers/record/7375445.

[16] B. Taha and A. Shoufan, "Machine Learning-Based Drone Detection and Classification: State-of-the-Art in Research," in IEEE Access, vol. 7, pp. 138669-138682, 2019, doi: 10.1109/ACCESS.2019.2942944.

[17] J. Kim, C. Park, J. Ahn, Y. Ko, J. Park and J. C. Gallagher, "Real-time UAV sound detection and analysis system," 2017 IEEE Sensors Applications Symposium (SAS), Glassboro, NJ, USA, 2017, pp. 1-5, doi: 10.1109/SAS.2017.7894058.

[18] S. Kümmeritz, "The Sound of Surveillance: Enhancing Machine Learning-Driven Drone Detection with Advanced Acoustic Augmentation," Drones, vol. 8, no. 3, p. 105, Mar. 2024. [Online]. Available: https://doi.org/10.3390/drones8030105. [Accessed: Aug. 2, 2024].

[19] X. Yue, Y. Liu, J. Wang, H. Song and H. Cao, "Software Defined Radio and Wireless Acoustic Networking for Amateur Drone Surveillance," in IEEE Communications Magazine, vol. 56, no. 4, pp. 90-97, April 2018, doi: 10.1109/MCOM.2018.1700423.

[20] M. Nijim and N. Mantrawadi, "Drone classification and identification system by phenome analysis using data mining techniques," 2016 IEEE Symposium on Technologies for Homeland Security (HST), Waltham, MA, USA, 2016, pp. 1-5, doi: 10.1109/THS.2016.7568949.