UAV DETECTION VIA DEEP LEARNING USING DATA FROM SMARTPHONE ACOUSTIC SENSORS

A FINAL REPORT SUBMITTED TO THE DEPARTMENT OF EARTH SCIENCES,
UNIVERSITY OF HAWAI'I AT MĀNOA, IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE
IN
EARTH AND PLANETARY SCIENCES

MARCH 2021
BY
JONATHAN TOBIN

COMMITTEE:
NEIL FRAZER (CHAIR)
MILTON GARCES
HENRIETTA DULAI

Abstract

Unmanned aerial vehicles (UAVs) have the potential to be used in actions antithetical to nuclear non-proliferation efforts. Therefore, the ability to effectively detect UAVs using ubiquitous hardware and software would be advantageous as both a deterrent and an initial warning system. Multi-rotor UAVs produce multiple sets of harmonics in the generated acoustics based on the rotational speed of each rotor. We look to exploit this unique characteristic to detect UAV presence through deep learning classification methods with audio data recorded on smartphones. Two flight tests using a DJI Matrice 600 (M600) UAV were conducted and recorded via eight Samsung Galaxy S8s. For periods with highest signal to noise ratio, the collected data were segmented and short-time Fourier transforms (STFTs) were applied to the waveforms. The resultant outputs were used to generate time-frequency representations to visualize the frequency changes over time. The same process was repeated for post-flight data when there was no UAV present. In addition, synthetic single harmonics were added to separate post-flight data. Real voice data was also recorded and used in training since speaking produces harmonics as well. To augment the dataset and increase the generalization of the model, Gaussian noise was added to each waveform and all spectrograms were regenerated. Data with UAV activity was given a label of one and data with no UAV activity was given a label of zero. Two convolutional neural network (CNN) binary classifiers were created and evaluated, one using the STFT outputs as input data and one using the spectrograms saved as images as input data. Both models performed well, with recall values of 0.963 and 0.986 and precision values of 1 and 1, respectively, on high SNR test data. Using test data with lower SNR, the models once again had precision values of 1 but recall values of 0.874 and 0.936, respectively. The maximum distance at which both models correctly predicted UAV activity was between 240 and 250 meters, though the image model was more consistent at longer distances. This work introduced two multi-rotor UAV detection models based on acoustic spectral characteristics and helps to validate the use of smartphones as acoustic sensors for UAV detection.

Table of Contents

1. Introduction

As unmanned aerial vehicles (UAVs) improve technologically and increase in public accessibility, their potential for use in actions antithetical to nuclear non-proliferation efforts increases as well. UAVs have the ability to transport and deliver external loads and perform reconnaissance activities, which if in the hands of bad actors may have grave consequences [1]. Whether being used nefariously or simply haphazardly, the ability to identify multi-rotor UAVs using ubiquitous, compact, and inexpensive hardware and software solutions would be advantageous for implementing an ad hoc detection system. This project explores the use of smartphones as acoustic sensors to record multi-rotor UAV flights and then utilizes deep learning to identify the presence of a UAV based on generated spectral characteristics in the recorded acoustic data.

Commercial off the shelf (COTS) mobile cyber-physical systems, such as smartphones, are ideal for creating lightweight and distributed detection networks. Smartphones have become ubiquitous sensor platforms with approximately four out of five adults in the United States reportedly owning one, and the total number steadily increasing [2]. Previous work has demonstrated the ability of certain smartphone microphones to be reliable acoustic sensors in both the audible and infrasonic sound ranges [3, 4]. Using smartphones as acoustic sensor platforms is advantageous not only because of the ubiquitous and compact nature of the devices but because of the evolving sophistication, which has enabled the development of edge processing techniques. In addition, the ability to enable a remote acoustic sensor or sensor network without having to necessarily deploy a specialized microphone or array of microphones would allow for rapid and far reaching sensor coverage. Since these are prominent COTS devices, there is continued opportunity for passive hardware improvement as smartphone upgrades are made by manufacturers. The increase is smartphone computational power and memory has enabled direct analysis of data at the source of collection and provided an avenue for directly displaying and/or communicating an output based on the analysis. In the context of this work, this would consist of detecting a UAV through acoustic data directly on the edge device and then displaying that message on the smartphone or sending it via cellular network or WIFI to a separate control station. This work begins this process by collecting acoustic data on smartphones and developing detection models to be implemented on the devices in future iterations.

While in operation, each UAV rotor (motor/propeller grouping) produces both broadband noise and harmonics [5]. In this work the harmonic patterns produced in the acoustics are the central characteristic utilized for detection. Each rotor produces a set of harmonics in its generated acoustics, where the fundamental frequency, or in this context the blade passage frequency (BPF), of these harmonics is dependent on the revolutions per minute (RPM) of the motor (*M_{RPM}*) and number of blades (*N_b*) of the propeller. This fundamental frequency is generally governed by the equation,

$$f_0 = M_{RPM} * N_b / 60$$

Since each rotor operates at a specific RPM, and assuming each rotor acts as an independent acoustic source while in motion, there may be from two to *N_r* number of rotors of unique harmonics identifiable. There are numerous scenarios in which the number of individual BPFs could range from two to *N_r*.

Depending on the UAV's flight velocity, equipment limitations, and environmental influences, while blade number will remain constant in flight, motor RPM may vary. To remain stable in air, a UAV must produce enough thrust from each rotor to counteract gravity but maintain a desired altitude and levelness. In the presence of wind, whether relatively continuous or more impulsive, in addition to compensating for gravity, the UAV must also compensate for these wind forces that act upon it. To counteract these forces and remain stable, the rotational speed of specific motors must be adjusted. As a motor's RPM varies, the fundamental frequency produced in its acoustics varies. Additionally, with increased air resistance the rotors would need to spin at higher speeds to counteract drag resulting in a higher fundamental frequency. During forward, reverse, or lateral motion, the back rotors (relative to direction of movement) operate at greater speeds than the front rotors so that there is increased thrust that propels the UAV in a certain direction while maintaining altitude. These clusters of rotors operating at targeted RPMs create clusters of identifiable harmonic patterns if the rotor speeds are stable or individual bands if speeds are inconsistent.

Recent work has been successful in detecting multi-rotor UAVs based on acoustic characteristics and through other physical phenomena and processing methods. Dumitretzki et al. developed a method that utilizes concurrent neural networks using Wigner-Ville spectrograms, Mel frequency cepstral coefficients (MFCCs), and mean instantaneous frequency classes in detecting UAVs and determining how many may be present [6]. Seo et al. utilized a convolutional neural network to detect the presence of a hovering UAV based on general spectral characteristics [7]. Sedunov et al., developed an algorithm to detect spectral patterns generated from UAVs, non UAVs, and noise based on harmonic patterns that does not rely a prior database. The algorithm then classifies the source of the pattern accordingly [8]. Bernardini et al., utilized a support vector machine framework for UAV binary classification that relies on MFCCs and a number of other temporal and spectral statistics as inputs [9]. Shi et al. proposed using a hidden Markov model to classify UAV activity in noisy environments based on MFCC feature extraction [10]. These methods relied on external microphone arrays or microphone systems with external processing units for analysis and eventual detection. Other machine learning techniques have also been employed to detect and classify UAVs, such as image recognition of the UAV itself [11, 12]. Fusing the methods of audio and visual detection have also been proposed for

UAV detection [13]. Radar techniques utilizing micro-doppler signatures have proven to be viable detection methods as well [14, 15].

This project utilized the acoustic time-frequency characteristics within a specific frequency range produced by the in-flight UAV rotors to evaluate if UAV activity was present or not present and at what distance from a smartphone detection occurred. This was accomplished by computing short-time Fourier transforms (STFTs) and generating spectrograms of recorded data and using these as inputs into convolutional neural networks. Two models were trained and evaluated, one utilizing the spectrograms saved and loaded as images and one using a transformed STFT output. These approaches capitalize on the multi-harmonic nature of the multi-rotor UAV. The specific UAV is not necessarily consequential, just that there are multiple harmonic bands within a specified frequency range.

2. Materials and Methods

2.1 Experimental Setup

The initial UAV flight experiment was conducted at Idaho National Laboratory's UAV runway. Eight Samsung Galaxy S8s were utilized as acoustic sensors, recording at 8kHz, and placed in protective cases directly on the ground. Each smartphone was positioned with the screen facing up (+z up), and with the top of the smartphone pointed north (+y north). The specific layout of the eight phones on the runway can be visualized below.

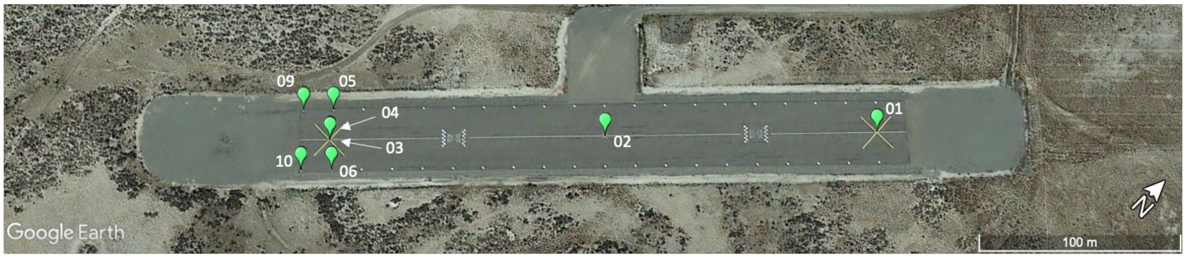

Figure 1. Smartphone layout for UAV test flights. Smartphone positions are indicated with green placemarks and numbers correspond to the station ID of each smartphone.

The UAV flight experiment utilized a DJI Matrice 600 (M600), which is a six rotor UAV (hexacopter). Specifically, the motor models were DJI 6010's and the propeller models were DJI 2170R's, which are double bladed propellers. The M600 was carrying an external load (GPS receiver) of approximately ten pounds, and all other system components remained stock. The external GPS attached to the UAV sampled at 1 Hz and provided ground truth to UAV position. Two flight tests, flight one and two, were conducted in succession with about a three minute gap between flights. The specific flight paths taken by the UAV during these two tests are as shown in **Figure 2**. Only data from specific sections of each flight were actually used, and the location of the UAV for these time segments is shown in red. For flight one, a total time of 7.83 minutes of flight time was used, and during these sections, the UAV mean speed was 2.330 m/s with a standard deviation (std) of 0.210 m/s. For the times used during flight two, the UAV had a mean speed of 2.474 m/s and std 2.108 m/s. The large standard deviation is due to the fact that for 4.30 out of the total 5.46 minutes of flight time used, the UAV had a mean speed of 1.585 m/s and with a small std 0.365 m/s, but for 0.33 minutes the mean speed was 8.512 m/s with a std 2.196 m/s, and for 0.83 minutes the mean speed was 4.817 m/s with a std of 1.621 m/s. Overall, the M600 flew relatively slowly given that the max speed is ~18 m/s in no wind [16].

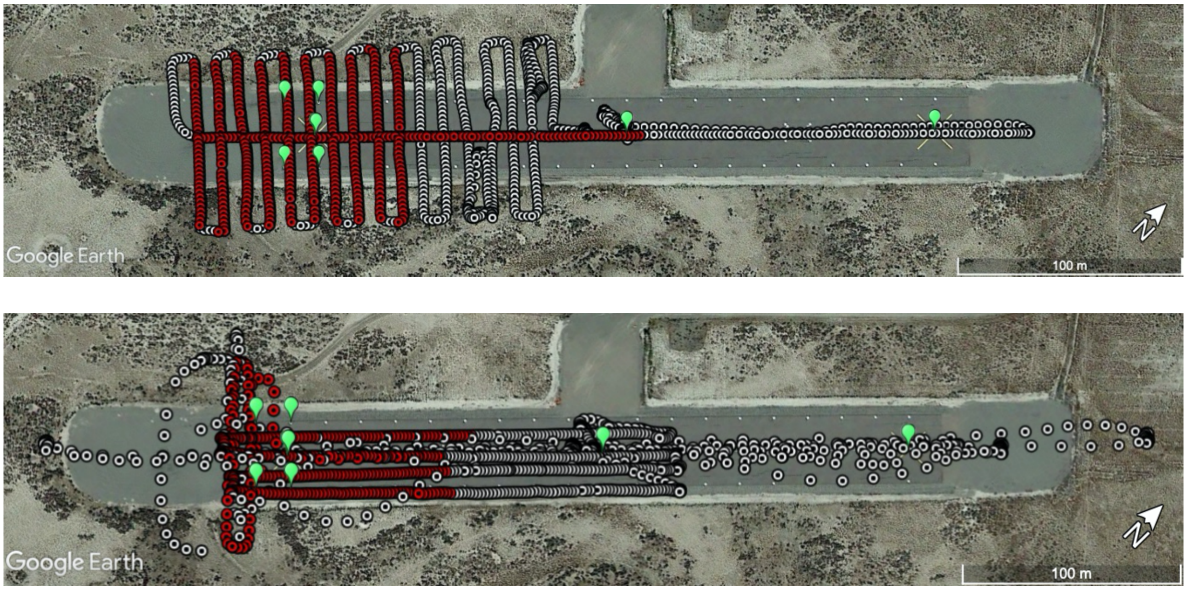

Figure 2. Flight path one **(top)** and flight path two **(bottom)** for M600 in two dimensions. White circles indicate the location of the UAV at a sampled time throughout the course of the flight and red circles indicate the position of the UAV at which recorded data is used in training. The green place-markers indicate the position of the Samsung Galaxy S8's.

2.2 Data Processing and Model Input

The recorded acoustic data was sectioned into ten second waveforms (total of 80k samples), with each successive waveform beginning and ending one second (8k samples) after the previous waveform. Ten second segments allow for adequately capturing the generated harmonic patterns when representing the spectral characteristics graphically. Sliding each waveform by one second allows for rapidly assessing new input data while maintaining the overall time-frequency representation, and augmenting the available dataset. As previously noted, only data from a subsection of each flight test was utilized. These data correlate to times with greater signal to noise ratio (SNR), where the targeted spectral characteristics are more discernable. The specific times selected were based off an empirical investigation of the collected dataset. In addition, only data recorded from phones 03-06 and 09-10 were utilized for training the models. These smartphones were clustered on one end of the UAV runway, evident in **Figure 1** above**.** For each ten second waveform, a short time Fourier transform (STFT) was computed and the output was used generate a time-frequency representation to visualize the spectral patterns in the waveform. The STFT performs successive discrete Fourier transforms (fast Fourier transforms in practice) of an input signal by truncating the signal into smaller sections multiplied by a windowing function [17]. The STFT is given by,

$$X[n, \omega) = \sum_{m=-\infty}^{\infty} x[n+m]\, w[m] e^{-j\omega m}$$

where *x[n]* is the input waveform, *w[m]* is the windowing function, and ω is frequency [17]. The output of the STFT, which is a matrix of complex STFT coefficients that represent the frequency content over a given time range of the input signal, is scaled by 2π divided by the size of windowed signal utilized in each Fourier transform. These values are then transformed to floating point bits (fbits) by taking the binary log of the absolute value of the STFT output array plus the 52 bit mantissa (IEEE standard 754 for 64 bit float) [18]. Transforming the STFT output into fbits is an alternative to decibel representation suitable for cyber-physical systems. Spectrograms were then generated to visually represent the frequency characteristics over the signal time duration.

The frequencies observable on the time-frequency representations were limited to 50 to 400 Hz, which appropriately captured the fundamental frequency, second, and third harmonic generated by each rotor. These lower frequency bands were chosen because, generally, lower frequency signals do not lose energy to the surrounding medium as rapidly as higher frequency signals. This will potentially enable detection at farther distances from the source than if concentrating on higher frequency bands. Also, as the harmonic order increases, the spatial variations in the spectral patterns increase, and the distinct frequency bands are not as clear. For the majority of the testing, the fundamental frequencies were centered at approximately 100 to 120 Hz. The following figure displays an example spectrogram for both in-flight and post-flight. The colormap of the spectrograms are normalized from the minimum to the maximum fbits value.

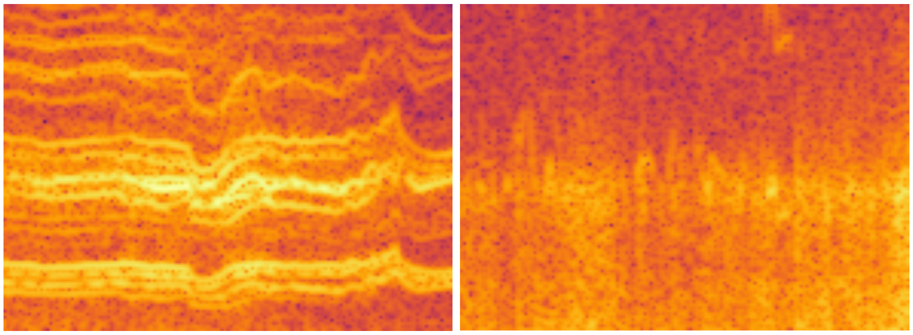

Figure 3. (left) Example spectrogram from flight two data with UAV activity. **(right)** Example spectrogram from post-flight with no UAV activity. The x-axis corresponds to time (total of 10s) and the unlabeled y-axis corresponds to frequency (50-400 Hz).

To both augment data for training and expand the scope of the model, Gaussian noise was added to each waveform and spectrograms were again generated for the noise corrupted signals. The mean of the noise was zero, and the standard deviation of the noise was the standard deviation of the waveform divided by 2^n, where n = 8. The figure below shows the noise corrupted spectrograms of the previous figure.

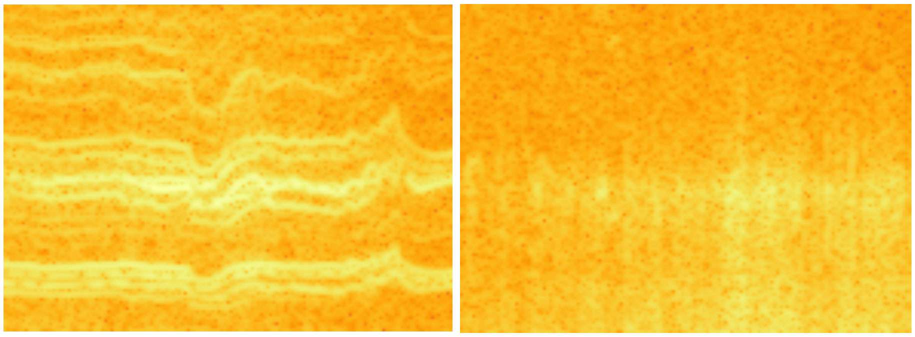

Figure 4. Example spectrograms with Gaussian noise added to each waveform. These spectrograms are generated from the same waveforms as shown in **Figure 3**.

Data from post-flight was also used as background noise to add synthetic harmonics. Using previously unused post-flight data, waveforms with linear and oscillating harmonics were added to the data. The fundamental frequency for each harmonic was generated randomly and was in the range 34 to 124 Hz (inclusive). Using this range of fundamental frequencies account for a variety of harmonic patterns visible between 50 and 400 Hz. This synthetic data was used in an attempt to train the model against waveforms with one harmonic pattern, such as a generator or fan would produce. The equation used to generate the synthetic waveform *x(t)* is described by the equation below. The constant a was set at 5 and the constant b was set at 0.5. The waveform was then multiplied by the mean of the average amplitude of the post-flight waveform for that specific instance.

$$x(t) = \sin\,(2\pi f_0 t + a\cos(2\pi bt))$$

Examples of the resulting spectrograms are shown below.

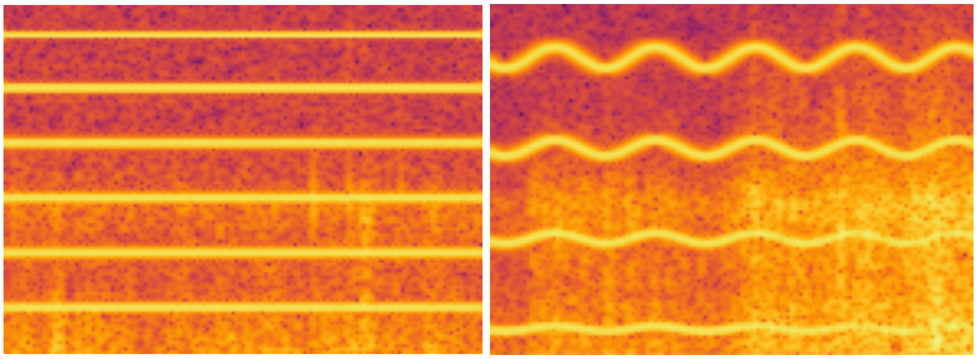

Figure 5. Example spectrograms of linear **(left)** and oscillating **(right)** synthetic harmonics added to post-flight data.

In addition to generating spectrograms of in-flight UAV, post-flight, and post-flight + synthetic data, spectrograms of real voice audio were also generated. The speech data was collected on a Samsung Galaxy S10 recording at 8kHz inside an office at the University of Hawaii at Manoa. The smartphone was positioned 30 cm down, 40 cm away, and 45 degrees right of the speaker's mouth, with the microphone pointed towards the speaker. Voice data was collected because of the harmonic nature of vowels in human speaking and the fundamental frequency of these harmonics generally fall into the frequency range examined here [19]. These data were utilized as another type of spectral pattern to train against.

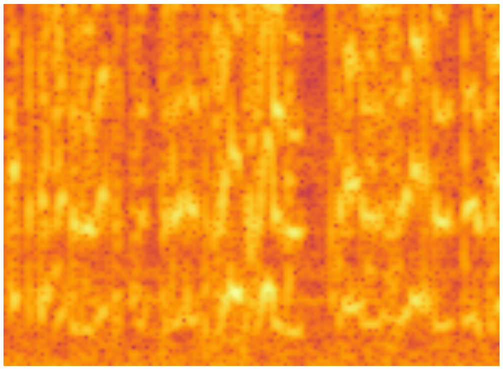

Figure 6. Example spectrogram of recorded vocals with visible harmonics.

The generated spectrograms were saved RGB images in the PNG file format and then cast into a four dimensional array, with array dimensions consisting of image number, pixels in x, pixels in y, and image channels. Each image had dimensions of 107x147, and since the images were saved in color, each image had three channels. The STFT fbits were also saved as an array with dimensions corresponding the total count, the number of frequency bands, which was 114, and the number of time bins, which was 626. Because the STFT generates frequency values in bins, the frequency range of fbits that were saved was from ~49 Hz to ~398 Hz. These arrays were used as the inputs for two independent convolutional neural network binary classifiers. The following table displays the total number of samples generated for training. Since image and fbits training data are equal in quantity, the table displays the counts for either type.

Table 1. Input counts for both the original and noise corrupted waveforms.

Waveform Origin Type	Original Waveform Count	Noise Corrupted Waveform Count	Subtotal	Total
Flight 1	2186	2186	4372	7956
Flight 2	1792	1792	3584	
Post Flight	936	936	1872	7294
Linear Synth	936	936	1872	
Wave Synth	935	935	1870	
Speaking	840	840	1680	

2.3 Model Architecture

A convolutional neural network (CNN) was selected as the learning framework for both models because this type of neural network has demonstrated the ability to extract complex patterns from images and temporally/spatially varying data, and then using these patterns to classify inputs [20, 21]. Since both image data and spectrogram data are interpreted based on these variations, a CNN model works well for both data types. A separate CNN model was created for the image and fbits inputs that utilized similar but slightly varying overall structure, layers, and hyperparameters. These are referred to as the image model and fbits model. The two flow charts below illustrate the model layers, and the input/output sizes of each layer may be observed.

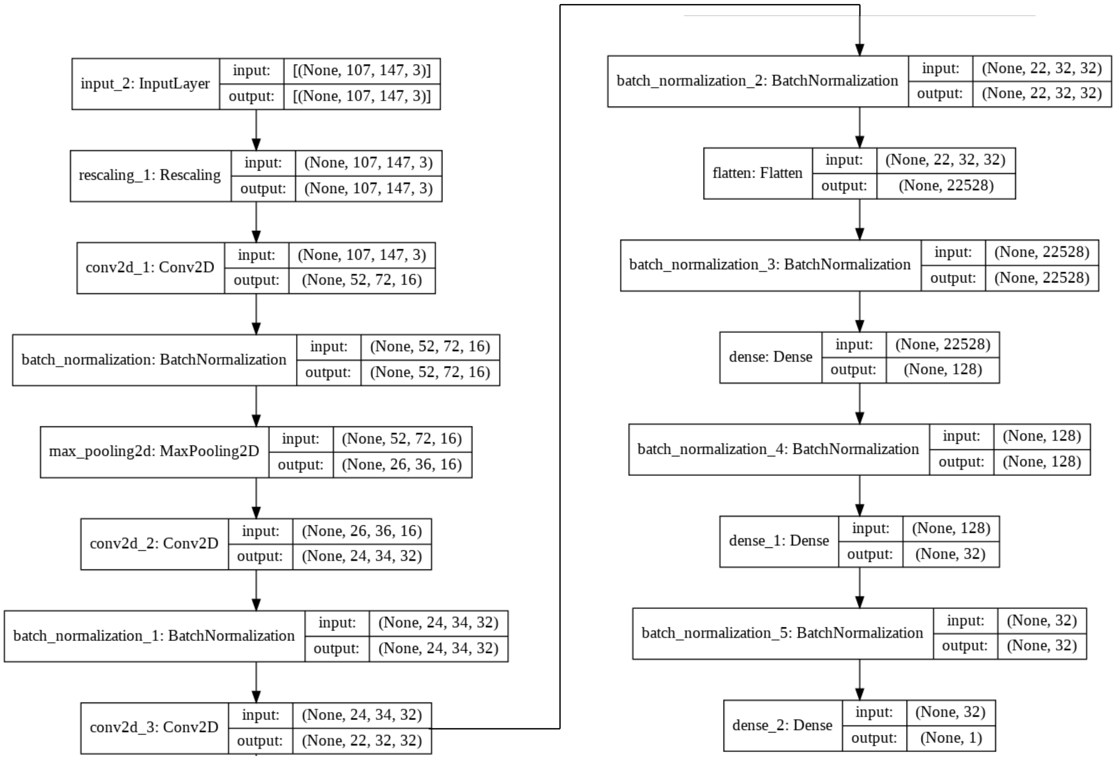
input_2: InputLayer
input: [(None, 107, 147, 3)]
output: [(None, 107, 147, 3)]
rescaling_1: Rescaling
input: (None, 107, 147, 3)
output: (None, 107, 147, 3)
conv2d_1: Conv2D
input: (None, 107, 147, 3)
output: (None, 52, 72, 16)
batch_normalization: BatchNormalization
input: (None, 52, 72, 16)
output: (None, 52, 72, 16)
max_pooling2d: MaxPooling2D
input: (None, 52, 72, 16)
output: (None, 26, 36, 16)
conv2d_2: Conv2D
input: (None, 26, 36, 16)
output: (None, 24, 34, 32)
batch_normalization_1: BatchNormalization
input: (None, 24, 34, 32)
output: (None, 24, 34, 32)
conv2d_3: Conv2D
input: (None, 24, 34, 32)
output: (None, 22, 32, 32)
batch_normalization_2: BatchNormalization
input: (None, 22, 32, 32)
output: (None, 22, 32, 32)
flatten: Flatten
input: (None, 22, 32, 32)
output: (None, 22528)
batch_normalization_3: BatchNormalization
input: (None, 22528)
output: (None, 22528)
dense: Dense
input: (None, 22528)
output: (None, 128)
batch_normalization_4: BatchNormalization
input: (None, 128)
output: (None, 128)
dense_1: Dense
input: (None, 128)
output: (None, 32)
batch_normalization_5: BatchNormalization
input: (None, 32)
output: (None, 32)
dense_2: Dense
input: (None, 32)
output: (None, 1)

Figure 7. Flow chart demonstrating CNN layer construction for image data inputs.

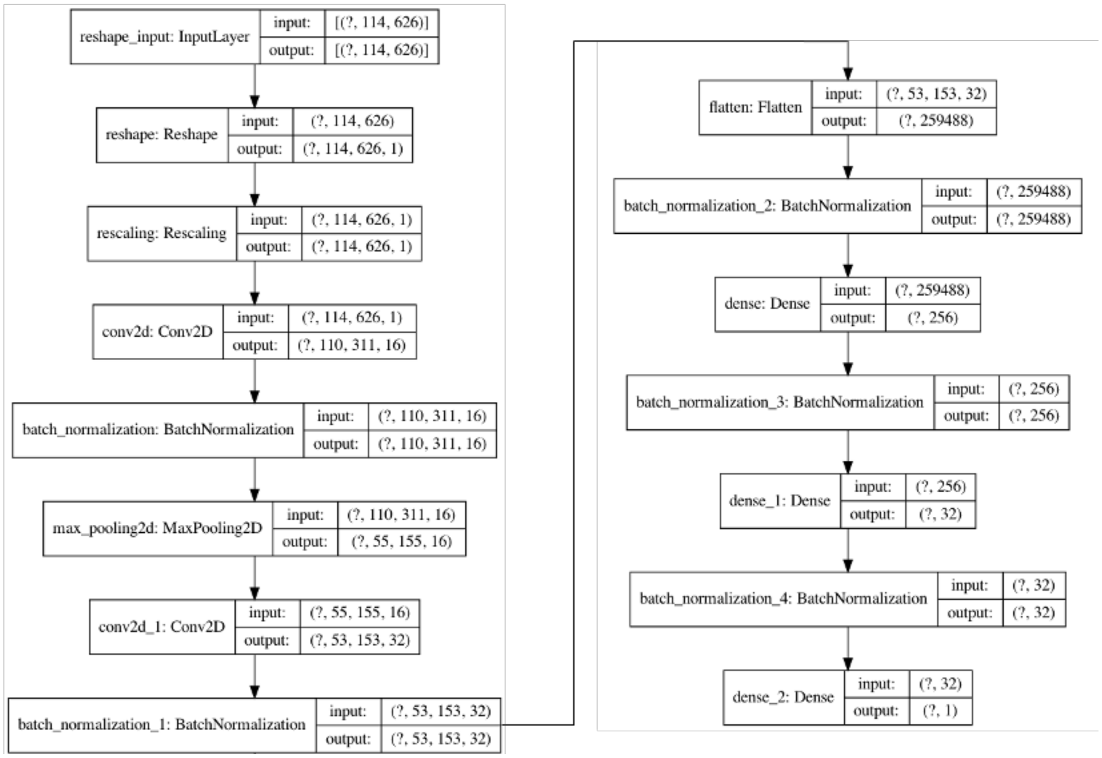

Figure 8. Flow chart demonstrating CNN layer construction for fbits data inputs.

These models, constructed using the TensorFlow Keras API v2.4.1, utilized two-dimensional convolutional and max pooling layers, a flattening layer, and fully connected layers [22]. The image model consisted of three two-dimensional convolutional layers with filter sizes of 16, 32, and 32, respectively. The first convolutional layers had a kernel size of 5x5 and a stride of 2, and the last two convolutional layers had kernel sizes of 3x3 with strides of 1. The first convolutional layer was followed by a two-dimensional max pooling layer with a pool size of 2x2. After the last convolutional layer there was a flattening layer, and three fully connected layers with output dimensionalities of 128, 32, and 1, respectively. The fbits model consisted of two two-dimensional convolutional layers with filter sizes of 16 and 32, respectively. The first convolutional layers had a kernel size of 5x5 and a stride of 1x2. This non-square stride was utilized because of the disparity between the x and y axes of the input fbits data. The second convolutional layer had a kernel sizes of 3x3 with a stride of 1. The first convolutional layer was followed by a two-dimensional max pooling layer with a pool size of 2x2. Since the input data shape varies between the two models, so too does the output dimensions of each convolutional layer. After the last convolutional layer there was a flattening layer, and three fully connected layers with output dimensionalities of 256, 32, and 1, respectively. For both models, batch normalization was used after activation of each layer, and the initial data was normalized from 0 to 1 after input. This normalization is built into the model so that any new data will be normalized appropriately and does not need to be normalized beforehand. A slight difference between the models was an initial reshaping layer at the beginning of the fbits model to add a channel dimension of one to the fbits input data. Both models utilized an adaptive moment estimation (Adam) optimizer with a learning rate set to 0.0003 for the image model and 0.0005 for the fbits model [23]. Since the CNNs are implemented as binary classifiers, either UAV activity (1) or no UAV activity (0), binary cross entropy was selected as the loss function, and is described by the equation,

$$L(X) = \frac{-1}{n}\sum_{i=1}^{n} y_{true}^{i}\log(y_{pred}^{i}) + (1 - y_{true}^{i})\ln(1 - y_{pred}^{i})$$

For all convolutional layers and the first two densely connected layers of each model, the rectified linear unit (ReLU) activation function was utilized. Both output layers used sigmoid activation so the output probability would be scaled between 0 and 1.

$$\text{ReLU: } f(x) = \begin{cases} 0\ for\ x \leq 0 \\ x\ for\ x > 0 \end{cases} \quad \text{Sigmoid: } \sigma(x) = \frac{1}{1+e^{-x}}$$

3. Results

Data recorded on phones 01 and 02 were used to evaluate the model. These phones had been placed about 280 m and 140 m from the larger cluster of phones, respectively. Using data from phones previously unused in training and that were located distant to the training cluster of phones provides the most unbiased evaluation of the model given the limited dataset. Specifically, STFTs and spectrograms were generated for recorded audio during flights one/two and post-flight. Synthetic harmonics were added to additional post-flight data, but with different fundamental frequencies and oscillation from what was used in training. The fundamental frequency range remained the same but the generated random numbers differed. For the oscillation, a was set to 4 and b was set to 0.58. Speech data was also recorded for a second time in different environmental conditions (outside on UH Manoa campus), but with the same distance from speaker to phone and phone orientation. Again, all STFTs and spectrograms were generated twice, once with the original waveforms and once with Gaussian noise corrupting the waveforms.

This process occurred twice over for a smaller and larger subset of the phone 01 and 02 data. First only high fidelity spectrograms with high SNR were selected for evaluation. The dataset was then expanded to include spectrograms with lower SNR, and ones which may have impulsive wind noise or other environmental noise corrupting the UAV acoustic signal. Evaluating the models on both subsets of data helps to illustrate the models' ability to predict on high quality data and to generalize to less pronounced signals in the spectrograms. The smaller test dataset had 1736 total samples and the larger test dataset had 4220 total samples, with each test set split evenly between 1 and 0 data labels (UAV/no UAV).

Normalized confusion matrices (NCMs) were generated to evaluate and compare the performances of the two models in prediction of the test data. The normalized confusion matrix displays the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) values from the predicted data as percentages of the actual class each belongs to. The normalized TP value is known as the true positive rate (TPR) or recall, and the normalized TN is known as the true negative rate (TNR) or specificity. The normalized FP is the false positive rate (FPR) and the normalized FN is the false negative rate (FNR). The precision metric (not displayed on NCM) assesses the accuracy of correctly labeling the positive class, in this instance, UAV activity. Equations for each of these values are described below.

$$TPR = \frac{TP}{TP + FN} \quad TNR = \frac{TN}{TN + FP} \quad FPR = \frac{FP}{TN + FP}$$

$$FNR = \frac{FN}{TP + FN} \quad precision = \frac{TP}{TP + FP}$$

These values are more appropriate and insightful in comparing binary classifiers than looking purely a single accuracy measurement. The NCMs for both the image model and fbits model using high SNR spectrograms are shown below. In predicting the presence of UAV activity, we assume that an output probability of <0.5 is a 0 (no UAV activity), and a probability >0.5 is a 1 (UAV activity).

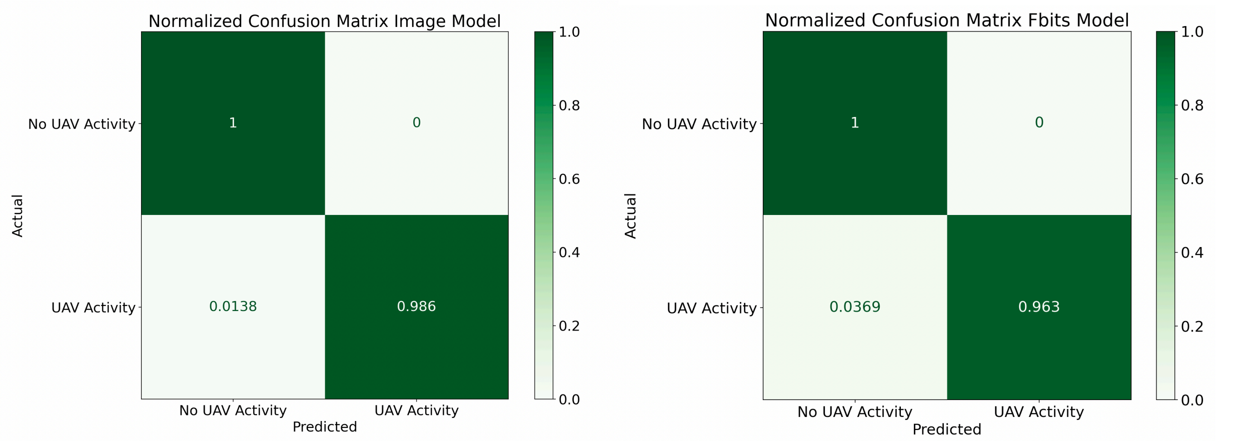

Figure 9. Normalized confusion matrices for image model **(left)** and fbits model **(right)** using sample data with high SNR. Counter-clockwise from the top right the boxes are normalized FP, TN, FN, TP.

Comparing the confusion matrices for both models, it is clear that for this dataset that both models were able to appropriately classify the no UAV activity data. This is evident from the TNR being 1 in both instances. However, the models do differ in their TPRs. The image model was slightly better at classifying UAV activity then the fbits model, with TPRs of 0.986 and 0.963, respectively. Because there were no FPs, the precision of each model is also 1.

The NCMs for the image model and fbits model now including lower SNR spectrograms comprising a larger test dataset are shown in the figure below.

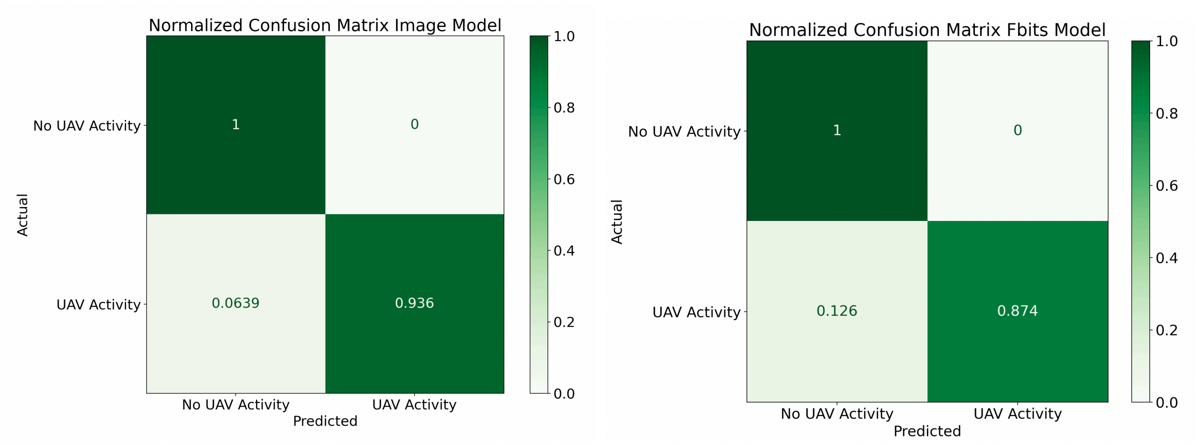

Figure 10. Normalized confusion matrices for image model **(left)** and fbits model **(right)** using larger sample data with lower SNR. Counter-clockwise from the top right the boxes are normalized FP, TN, FN, TP.

Even with a larger test set, both models we're able to appropriately label the zero data – there were no FPs and both had precisions of 1. This time however, the recall for both models was lesser. Again, the image model performed better than the fbits model, with TPRs of 0.936 and 0.874, respectively. The TRP of the image model dropped by 0.05 when the test set was expanded, and the TPR of the fbits model decreased by 0.089.

The harmonic mean of precision and recall is defined by the F_1 score. A larger F_1 score indicates that precision and recall are both high.

$$F_1 = \frac{TP}{TP + \frac{FN + FP}{2}} = \frac{2}{(precision)^{-1} + (recall)^{-1}}$$

Looking just at the larger test set, the F_1 score for the image model was 0.967 and the F_1 score for the fbits model was 0.933. This difference was as expected since the precision was the same for both models, but the image model had a higher recall value.

An important metric for detection in this context is the distance from the smartphone at which the model can detect the presence of a UAV. This is highly dependent on acoustic power produced from the rotor rotation and the environmental noise conditions impacting the SNR on the spectrogram. For both models, using original waveforms, the maximum distance at which the UAV was detected was in the range of 240-250 m. The histograms below display the frequency of correctly labeled UAV predictions against total true count for distance ranges during flight two for phones 01 and 02. This only examines data from the original, non-noise corrupted waveforms. The distance refers to the calculated two dimensional distance in meters between the moving UAV's latitude/longitude coordinates and the stationary phones latitude/longitude coordinates. The range for UAV to phone 01 distances is 0 to 350 m and the range for UAV to phone 02 distance is 0 to 250 m. Each plot uses 10 m bin ranges.

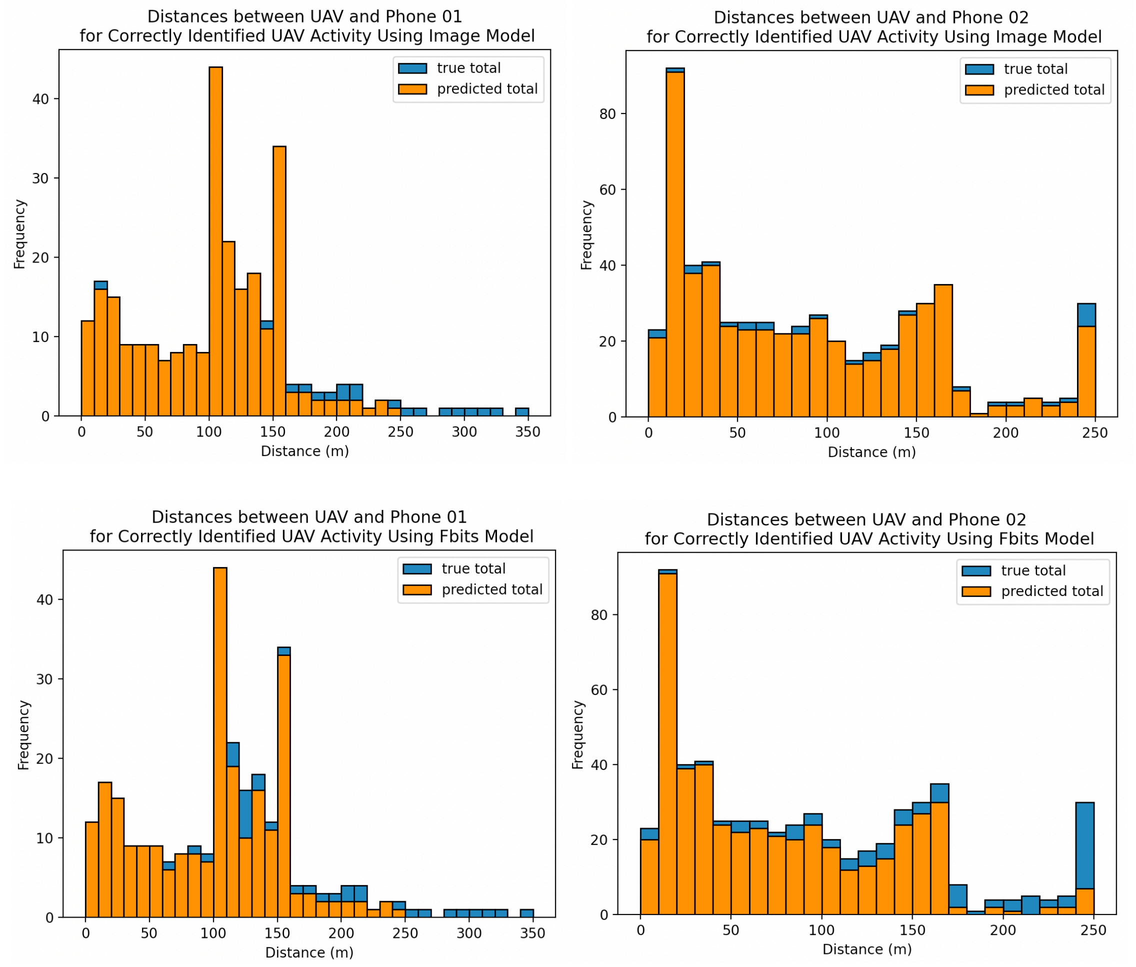

Figure 11. Histograms of comparing positive prediction distances and true totals of flight two data recorded on phones 01 **(left top/bottom)** and 02 **(right top/bottom)** for both models.

Looking at the flight two data for phone 01, from 0 through 110 m, both models perfromed well, with the fbits model only incorrectly predicting three samples and the image model only incorrectly predicting one sample. None of the missclassified samples overlapped between the models. From 110 through 160 m the image model correctly classified all but one sample, however, the fbits model faired poorer, missclassifying 13 of the total 102. About half of these missclassifications came from the

120 to 130 m range. From 160 m through 350 m, both models were even in their predictions, and both models were unable to correctly classify any data greater than the 240 to 250 m range. At this range data was sparse. For the flight two data from phone 02, both models performed well thorugh 170 m, with the image model predicting slightly better, particularily from 110 m to 170 m. This trend is also conistent with the phone 01 data in that distance range. From 170 to 250 m, the image model performed substantially better than the fbits model, correctly labeling 50 of the total 61, where the fbits model only correctly labeled 16. Again both models did correctly label data at the distance range of 240 to 250 m, but the image model was more correctly consistent at greater distances.

4. Discussion

Two new convolutional neural network models are proposed for UAV binary detection based on acoustic harmonic patterns of rotors recorded from smartphone acoustic sensors. One model utilizes images of spectrograms and one model utilizes the normalized fbits from the transformed output of the STFT. Both the image and fbits CNN models proved to be viable for UAV binary detection, with the image model outperforming the fbits model. Both models correctly identified UAV activity at maximum distance ranges of 240 to 250 m, though the image model had greater overall accuracy and consistency at that range. The fbits model is also an initial use of fbits in the context of deep learning and time-frequency UAV detection methods. In the context high speed detection and edge processing on smartphones, or similarly low-cost COTS devices, the fbits model implementation would be more ideal because it requires fewer processing steps to complete the same evaluation when compared to the image model. The fbits model necessitates only STFTs and band-passing the specified frequency range, while the image model would require spectrogram generation, band-passing to the specified frequency range, and saving and loading the spectrogram as an image before inference/prediction.

This project has also demonstrated the viability of utilizing smartphones for data collection of a moving UAV source and building a curated training dataset from the collection. With smartphone applications such as the RedVox app, developed by M. A. Garces and RedVox, which allows for real time acoustic collection utilizing the built-in microphone of a smartphone on which application is installed, it would be possible to port a condensed model into the application for real time inference and transmission of detection output [24]. The next steps of this project include preparing the models for deployment into smartphones and other edge devices. This work also annotated a unique acoustic dataset recorded on smartphones of an in-flight M600 flying at relatively slow speed and at a low altitude (relative to ground). Publicly available UAV acoustic datasets are very limited, and this will help increase dataset quantity and accessibility.

Additional multi-rotor UAV flight tests are planned to increase the available acoustic dataset and test against the current models/develop more robust models. These additional tests will vary in flight path and altitude from the original two flight tests examined in this project. The same UAV will be utilized, again carrying an external GPS system. The new tests will be conducted at the same location, but the smartphones will be placed in different positions. The flight test will utilize nine Samsung Galaxy S8 smartphones, arranged in the orientation seen below and labeled from 01 to 09.

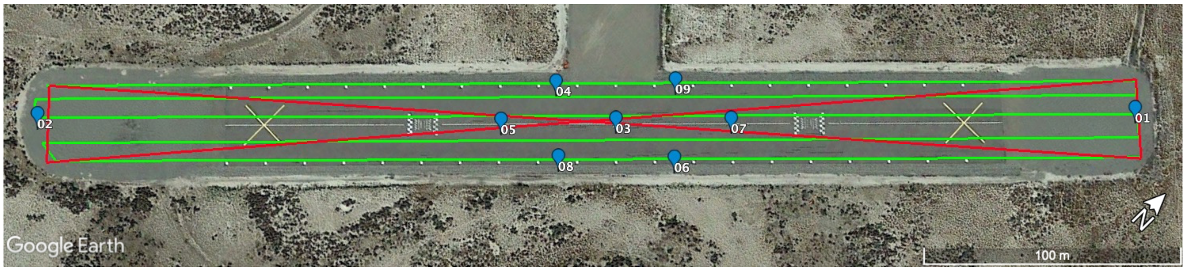

Figure 12. New Smartphone Layout with flight paths.

The smartphones will once again be placed in protective cases directly on the ground, with screens facing towards the sky and with the top of the smartphone pointed north. The devices will record at 8kHz. This new orientation of phones serves multiple purposes. First, placing two phones at opposite ends of the runway, in the figure phones 02 and 01 (left and right, respectively), will maximize the distance between a phone and the UAV while the UAV is at the opposite end. The distance between 02 and 01 is ~430m.

This allows for an expanded range to both test the detection distance capabilities of current methods, and collect data at greater ranges. The positioning of the centered smartphones (03-09) will too capture the UAV at various (shorter maximum) distances, but the staggered positioning of the devices aims to assist in the ability to estimate UAV position.

There will be two distinct flight patterns, displayed in figure above in green and in red, which the UAV will follow. The UAV will first fly the green path followed immediately by the red path without landing/hovering. These flights will be repeated five times at targeted altitudes of 5, 10, 25, 50, and 100m. For each iteration, the UAV will attempt to maintain an overall speed of 10m/s. The phones will begin recording at least 10 minutes before the initial flight, and between each change in altitude thereafter there will be a 5 minute wait (though devices will continue recording throughout duration). After the last flight, the phones will record for an additional 10 minutes before stopping.

5. Acknowledgements

I would like to thank my advisor Milton Garces for providing me the opportunity to work on this project and for the guidance along the way. I would like to thank our Idaho National Laboratory Collaborators, David Chichester, Scott Watson, and Jay Hix, for conducting the physical experiments and providing the necessary equipment and venue.

I would also like to thank my other committee members, Neil Frazer and Henrietta Dulai, for taking an interest in my work and providing valuable feedback.

6. References

[1] Yaacoub, J. P., Noura, H., Salman, O., & Chehab, A. (2020). Security analysis of drones systems: Attacks, limitations, and recommendations. *Internet of Things*, *11*, 100218. https://doi.org/10.1016/j.iot.2020.100218

[2] Demographics of mobile device ownership and adoption in the United States. (2020, June 05). Retrieved March 3, 2021, from https://www.pewresearch.org/internet/fact-sheet/mobile/

[3] Asmar, K., Garcés, M., & Williams, B. (2019). A method for estimating the amplitude response of smartphone built-in microphone sensors below 4 kHz. *The Journal of the Acoustical Society of America*, *146*(1), 172, https://doi.org/10.1121/1.5110723

[4] Asmar, K., Garcés, M., Hart, D., & Williams, B. (2018). Digital acoustic sensor performance across the infrasound range in non-isolated conditions. *The Journal of the Acoustical Society of America*, *144*(5), 3036. https://doi.org/10.1121/1.5078591

[5] Roger M, Moreau S (2020). Tonal-Noise Assessment of Quadrotor-Type UAV Using Source-Mode Expansions. *Acoustics*. 2(3):674-690. https://doi.org/10.3390/acoustics2030036

[6] Dumitrescu, C. et al. (2020). Development of an Acoustic System for UAV Detection. *Sensors (Basel, Switzerland)* vol. 20,17 4870. https://doi.org/10.3390/s20174870

[7] Seo, Y., Jang, B., & Im, S. (2018). Drone Detection Using Convolutional Neural Networks with Acoustic STFT Features. *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS),* Auckland, New Zealand. https://doi.org/10.1109/AVSS.2018.8639425

[8] Sedunov, A., Haddad D., Salloum, H., Sutin, A., Sedunov, N. & Yakubovskiy, A. (2019). Stevens Drone Detection Acoustic System and Experiments in Acoustics UAV Tracking. *2019 IEEE International Symposium on Technologies for Homeland Security (HST)*, Woburn, MA. https://doi.org/10.1109/HST47167.2019.9032916

[9] Bernardini, A., Mangiatordi, F., Pallotti, E., & Capodiferro, L. (2017). Drone detection by acoustic signature identification. *Society for Imaging Science and Technology.* https://doi.org/10.2352/ISSN.2470-1173.2017.10.IMAWM-168

[10] Shi L., Ahmad I., He Y. & Chang K. (2018). Hidden Markov model based drone sound recognition using MFCC technique in practical noisy environments. *Journal of Communications and Networks*. 20. 509-518. https://doi.org/10.1109/JCN.2018.000075

[11] Coluccia, A. et al. (2019). Drone-vs-Bird Detection Challenge at IEEE AVSS2019. *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS),* Taipei, Taiwan. https://doi.org/10.1109/AVSS.2019.8909876

[12] Unlu, E., Zenou, E., Riviere, N. et al. (2019). Deep learning-based strategies for the detection and tracking of drones using several cameras. *IPSJ T Comput Vis Appl 11*. https://doi.org/10.1186/s41074-019-0059-x

[13] Jamil, S., Fawad, Rahman, M., Ullah, A., Badnava, S., Forsat, M., Mirjavadi, SS. (2020). Malicious UAV Detection Using Integrated Audio and Visual Features for Public Safety Applications. *Sensors*. https://doi.org/10.3390/s20143923

[14] Rahman, S., & Robertson, D. A. (2018). Radar micro-Doppler signatures of drones and birds at K-band and W-band. *Scientific reports*. https://doi.org/10.1038/s41598-018-35880-9

[15] Hoffmann, F., Ritchie, M., Fioranelli, F., Charlish, A., & Griffiths, H. (2016). Micro-Doppler based detection and tracking of UAVs with multistatic radar. *2016 IEEE Radar Conference (RadarConf)*, Philadelphia, PA. https://doi.org/10.1109/RADAR.2016.7485236

[16] Matrice 600 - DJI. (n.d.). Retrieved from https://www.dji.com/matrice600

[17] Oppenheim, Alan V., Ronald W. Schafer, John R. Buck "Discrete-Time Signal Processing", Prentice Hall, 1999

[18] Garcés, MA. (2020). Quantized Constant-Q Gabor Atoms for Sparse Binary Representations of Cyber-Physical Signatures. *Entropy*. https://doi.org/10.3390/e22090936

[19] Palmer A.R., Winter I.M. (1993) Coding of the Fundamental Frequency of Voiced Speech Sounds and Harmonic Complexes in the Cochlear Nerve and Ventral Cochlear Nucleus. In: Merchán M.A., Juiz J.M., Godfrey D.A., Mugnaini E. (eds) The Mammalian Cochlear Nuclei. NATO ASI series (Series A, Life sciences), vol 239. Springer, Boston, MA. https://doi.org/10.1007/978-1-4615-2932-3_29

[20] O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. *ArXiv,* https://arxiv.org/*abs/1511.08458*

[21] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn and D. Yu (2014). Convolutional Neural Networks for Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing,* vol. 22, no. 10, pp. 1533-1545. https://doi.org/10.1109/TASLP.2014.2339736

[22] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale

machine learning on heterogeneous systems, 2015. Software available from tensorflow.org

[23] Kingma, D. P., & Ba, J. (2017). Adam: A Method for Stochastic Optimization. https://arxiv.org/abs/1412.6980

[24] Garces, M. A. (n.d.). App. Retrieved from https://www.redvoxsound.com/app.html