Estimation of number of unmanned aerial vehicles in a scene utilizing acoustic signatures and machine learning

Wilson A. N., Ajit Jha, Abhinav Kumar, Linga Reddy Cenkeramaddi

Abstract: **With the exponential growth in unmanned aerial vehicle (UAV)-based applications, there is a need to ensure safe and secure operations. From a security perspective, detecting and localizing intruder UAVs is still a challenge. It is even more challenging to accurately estimate the number of intruder UAVs on the scene. In this work, we propose a simple acoustic-based technique to detect and estimate the number of UAVs. Our method utilizes acoustic signals generated from the motion of UAV motors and propellers. Acoustic signals are captured by flying an arbitrary number of 10 UAVs in different combinations in an indoor setting. The recorded acoustic signals are trimmed, processed, and arranged to create a UAV audio dataset. The UAV audio dataset is subjected to time-frequency transformations to generate audio spectrogram images. The generated spectrogram images are then fed to a custom lightweight convolutional neural network (CNN) architecture to estimate the number of UAVs in the scene. Following training, the proposed model achieves an average test accuracy of 93.33% as compared to state-of-the-art benchmark models. Furthermore, the deployment feasibility of the proposed model is validated by running inference time calculations on edge computing devices such as the Raspberry Pi 4, NVIDIA Jetson Nano, and NVIDIA Jetson AGX Xavier.**

D.1 Introduction

Advancements in chip miniaturization and wireless connectivity have made unmanned aerial vehicle (UAV) based solutions attractive in various applications such as agriculture [1], disaster management [2], aerospace [3], law enforcement [4], etc. Their widespread popularity can be attributed to their unparalleled maneuverability, decreasing cost, and increased sophistication [5]. However, the advantages of UAVs are also exploited for improper and illegal use [6]. Further, as UAVs are compact and small in size, concerns regarding collisions with other air-borne entities, privacy, security, delivery of dangerous payloads, etc. should also be addressed.

Ground control stations should be well-equipped with systems that can detect and monitor UAV activity based on requirements.

There have been multiple attempts throughout the literature pertaining to the detection and tracking of small-size UAVs. Some of these approaches include the use of WiFi signals, RF radiations, vision-based sensors, radar-based approaches, and acoustic signatures [7], [8]. However, compared to other sensors, utilizing acoustic sensors for UAV detection has been shown to exhibit a number of advantages. Acoustic sensors are low-cost compact devices that detect pressure fluctuations created by sound waves. Unlike traditional vision and radar-based sensors, acoustic sensors are typically omnidirectional in nature. This allows them to sense disturbances in all directions making them an ideal choice for collision-avoidance systems. Further, passive acoustic sensors do not emit any radiation and hence it is environment friendly. As acoustic signals are independent of the UAV form factor, it fairs well in comparison to radar systems that require a threshold radar cross-sectional area to enable detection. The output data rate for acoustic sensors is comparatively low as compared to vision and radar-based systems facilitating seamless data acquisition and processing. However, the limitation of acoustic systems is their low detection range. For small inexpensive acoustic sensors, the detection range is typically less than 300 m facilitating only short-range detection [9]. In this effort, we address the problem of detecting and estimating multiple UAVs in an indoor environment using acoustic sensors and machine learning techniques.

D.2 Related Work

In this section, we provide a brief overview of the various works in the literature pertaining to UAV detection using acoustic signatures. It was observed that small multi-rotor UAVs can produce complex acoustic fields due to UAV motorization and propeller motion. The acoustic field thus produced contains complex harmonics and is arguably a unique characteristic of the UAV. Investigations related to the study of the UAV acoustic field can be found [10], [11], [12] in the literature. In [10] investigated the effect of UAV acoustics on human subjects in an indoor occupational environment. They concluded that the ability to accurately identify and record UAV acoustic signatures primarily depends upon the sound pressure level of the UAV. The study also found that an efficient redesign of multi-rotor UAVs is essential to lower noise levels and regulate the noise frequency spectrum produced by UAVs. A similar study [11] utilizes a large-aperture scanning microphone array to measure the sound pressure level of a hovering UAV. The obtained results are then used to determine a better design of UAV acoustics and analyze the UAV acoustic field. In [12], the authors utilize the experimental data obtained from [11] to model the sound pressure level of a UAV acoustic field using a physics-infused machine learning algorithm. The physics-infused model is developed utilizing the interference from sound pressure waves that are produced from acoustic monopole sources. The above-mentioned works indicate that UAV acoustic signatures are unique and depend on a number of factors. Our work utilizes these unique acoustic harmonic signatures generated by multiple UAVs to detect and estimate their presence in an indoor setting. In the following subsections we provide a brief account of the various techniques used for UAV detection based on acoustic signatures.

D.2.1 Conventional approach

The conventional approach relies on correlation and other signal-processing techniques to detect UAVs from acoustic signatures. One early approach by [13], uses a stationary microphone array to detect and track the flight trajectory of low-altitude UAVs using their engine sound. UAV detection is performed by exploiting the Doppler shift of the engine sound. On the other hand, tracking is carried out by estimating the direction of arrival of the UAV. The direction of arrival is obtained by utilizing the acoustic phase shift over the microphone array. The real-time performance of the proposed approach was evaluated by performing field experiments. It was found that the propagation delay of the acoustic delay impacts the UAV detection and tracking performance. Another approach in [14], utilized a modified cross-correlation technique for UAV detection. The proposed approach leverages the differential Doppler shift that is created due to the high-speed UAV motion and microphone array separation. Using the differential Doppler shift property, the received signals are successfully decorrelated from the ambient noise to enable UAV detection. Measurements were carried out in a controlled area with little noise. Ambient noise was later added to the obtained data to evaluate the performance. Another work [9] for UAV detection and classification involved the development of a drone acoustic detection system (DADS) using microphone nodes. In the DADS, the detection of UAVs is performed using the steered-response phase transform (SRP-PHAT) method while classification is obtained by utilizing the propeller frequencies from the spectrogram of the measured acoustic signatures. It is reported that the SRP-PHAT method provides reliable performance with real UAVs in real-world scenarios. However, the classification algorithm requires the UAVs to obtain a threshold distance to achieve better performance. The work by [15], utilizes the Barlett, Capon, and cross-correlation method to study and analyze the acoustic spectrum generated from UAVs. The proposed approach utilizes additional high-pass filters to obtain performance similar to the mel-frequency cepstral coefficients (MFCC) method for UAV detection. Further experiments also reveal that the cross-correlation method exhibited superior performance when followed by low-pass filtering to remove noise. In comparison, [16], utilizes the intrinsic harmonics inherent in the acoustic field signatures for detection and 3D localization. The acoustic signal's fundamental frequency and a few relevant harmonics are extracted using a pitch detection algorithm coupled with zero-phase bandpass filters. Experimental measurements have been carried out in anechoic and outdoor environments for performance evaluation. It was observed that the proposed approach fairs well when the UAV traverses in simple vertical trajectories. However, for complex trajectories with multiple UAVs, the performance was low and required additional research. The study by [17] also provides insights into utilizing acoustic information to detect UAVs. In this work, the authors measure the noise level using a sound level meter in a controlled environment containing only one UAV. After obtaining the noise-free measurements, anthropogenic noise of people and background music was generated in the environment along with the UAV. Spectral methods were used to analyze the frequency spectrum and it was reported that UAV detection was confirmed upon observing a 5000 Hz frequency in the spectrum. The research however fails to reproduce and verify the results in a real-world environment with multiple UAVs. In [18], the authors have addressed the problem of UAV detection through a biologically inspired vision approach. The spectrogram signals obtained after performing time-frequency analysis on the audio signals provide meaningful information that is embedded in noise. By preprocessing these spectrogram images through a hoverfly vision model, useful representations of these audio signals can be retrieved. The extracted representations can be utilized for UAV detection. It was shown through outdoor field trials that the bioinspired technique can improve the maximum UAV detectable distance between 30% and 50% with respect to traditional narrowband and broadband techniques. However, the proposed approach requires additional verification by using more UAV experiments and flight scenarios.

D.2.2 Machine learning approach

The advent of machine learning has bought new capabilities for UAV detection and classification [19], [20]. Machine learning techniques identify the inherent hidden patterns from the data that aid in UAV detection and classification [21]. By utilizing these techniques with additional preprocessing techniques such as short-time Fourier transform (STFT), principal component analysis (PCA), etc., a significant improvement in detection and classification accuracy has been reported [22]. In the subsequent sections, we explore some of the existing literature that uses machine learning techniques for UAV detection and classification in terms of unsupervised and supervised learning algorithms.

D.2.2.1 Unsupervised learning

In unsupervised learning [23], the algorithm learns to extract the hidden patterns from the data. These algorithms work primarily on unlabeled data and learn the inherent structure of the data without the need for any human intervention. They are mainly used for tasks such as clustering, association, and dimensionality reduction. In [24], the authors study the acoustic fields generated by various small quadcopter UAVs. The data thus obtained is used along with simulation software such as COMSOL Multiphysics to perform numerical simulations and analysis. The study determined the influence of blade defects, directional patterns, and pressure variations caused by UAV propellers on UAV acoustic fields. The collected acoustic signatures were further provided to a neural network that is trained on the cepstrum coefficients to obtain UAV detection. In [25] the authors used multiple microphone nodes to detect and track a UAV in a real-world environment with background noise.

The work used MFCC and STFT for preprocessing the data. This was followed by using support vector machines (SVM) and convolutional neural networks (CNN) for training. Reported results indicate that the STFT-SVM model exhibited better performance to detect a single UAV when the UAV approaches the vicinity of a microphone node. Future work can include experimenting with multiple UAVs and also reducing background noise during preprocessing. The work in [26] utilizes a multi-class SVM for identifying UAVs in diverse environmental conditions. A dataset is created that contains five 70 minute audio from nature during the daytime, street traffic, train, crowd, and flying UAVs. The audio files are trimmed to 5 second and 20 millisecond segments for analysis. Preprocessing is then performed to extract temporal centroid, spectral roll-off, spectral centroid, zero crossing rate (ZCR), MFCCs, etc., as features. The extracted features are then fed to the SVM classifier to obtain a high UAV detection accuracy of 96.4%. In [27], the authors perform UAV detection using two classifiers, plotted image learning (PIL) and k nearest neighbors (KNN). Sound clips with a 1 second duration of DJI Phantom 1 and 2 are recorded separately both indoors and outdoors in a noise-free environment. Later, outdoor environment sounds are added to simulate real-world scenarios. The FFT is applied to the sound clips which are then fed to the different classifiers. The authors reported that PIL showed 83% accuracy in UAV detection as compared to KNN which accounted for 61%. In [28], the authors developed a distributed system using acoustic wireless sensor network for UAV detection and localization. Through trial experiments, it was observed that the power spectral density (PSD) of UAV sound differed significantly from the background spectrum. On the basis of this concept, an acoustic dataset was created. The dataset consisted of UAV sounds that were augmented with background environment sounds. The sound clips are low pass filtered at 15 kHz, after which the PSD is obtained using FFT. PCA is further performed for dimension reduction. The preprocessed signals are then divided for training, testing, and additional testing with overlapped signals and subsequently fed to the SVM classifier. It was reported that UAV detection was successful when the introduced signal-to-interference ratio (SIR) was greater than 10 dB. In this work [29], the authors use the blind source separation (BSS) method to detect UAVs in the presence of multiple source interference. Three different UAVs are used separately to capture the audio signatures. The proposed method works by first estimating the number of sources. After source estimation, three methods ICA, PCA, or variational mode decomposition (VMD) are applied based on the type of source separation (overdetermined, positive-definite, or underdetermined) required. The features extracted are then fed to different machine learning algorithms such as SVM, KNN, and decision trees to evaluate the performance. It was reported that SVM and KNN showed similar performance with SVM exhibiting slightly better performance. Both algorithms exhibited an accuracy of more than 90% for UAV detection outperforming traditional filtering and mixed-signal methods. Another approach for acoustic-based UAV detection was performed by [22]. In this work, the sounds of amateur UAVs, birds, airplanes, and thunderstorms are recorded in a noisy environment. The authors use MFCC and linear predictive cepstral coefficients

(LPCC) for feature extraction. The extracted features are then fed to SVMs with linear, cubic, and quadratic kernels to detect and identify UAV acoustics. Results show that SVM with the cubic kernel when coupled with MFCC features outperformed LPCC with a UAV detection accuracy of around 96.7%.

D.2.2.2 Supervised learning

In contrast to unsupervised learning, supervised learning [23] utilizes labeled datasets as inputs and outputs. The labeled data serves as a kind of supervision to help the model learn the structure of the data. Supervised algorithms can learn over time and improve its accuracy based on the amount of labeled data and its inherent structure. In [30], the authors perform a comparative study to determine the best classifier for acoustic UAV detection. Acoustic signatures from various UAVs are recorded individually and augmented with diverse environmental noise to simulate real-world UAV scenarios. The MFCCs from these signals are extracted and fed to different classifiers and their performance is evaluated. It was reported that RNN provided the best performance with an F-score of 80%, followed by the Gaussian mixture model (GMM) with 68% and CNN at 58%. The study in [31] uses normalized STFT on UAV acoustic signals. The UAV acoustic signatures are recorded using DJI Phantom 3 or 4 models. The recorded sound clips are trimmed to a length of 20 ms with 50% overlapping. Normalized STFT is performed on these sound clips to obtain 41958 non-UAV and 68931 UAV sound frames. The non-UAV sound frames contained acoustic signatures from scooters and motorcycles. The output obtained after performing the STFT is then fed to the proposed CNN architecture after adding additive white Gaussian noise (AWGN). Results reported a 98.97% UAV detection accuracy and 1.28 false alarm rate (FAR). In this work [32], the sensory substitution of pre-existing and ambient microphones along with CNN is used to detect remotely piloted aircraft systems (RPAS) in urban environments. Indoor and outdoor experiments were carried out individually with a diverse set of RPASs. Spectrogram images are generated from the recorded audio clips. The spectrogram images are then further used to train the Inception CNN model via transfer learning. Results showed an RPAS detection accuracy of greater than 90% for all RPAS classes. The work in [33] uses mel-spectrograms to extract the features from the audio signals of UAVs. The extracted features are then used with CNNs and CRNNs for UAV classification. It was concluded that CNNs exhibited superior performance in the classification of UAVs from the obtained mel-spectrograms. Further, the study also investigated the use of late fusion methods with ensemble techniques to improve UAV detection performance. Another work by [34] also utilizes the audio spectrograms along with CNN, recurrent neural network (RNN), and convolutional recurrent neural network (CRNN) to identify and detect UAVs. The authors conducted two experiments using two different UAVs in a controlled environment. Real-world background noises were augmented to obtain realistic audio information that can be used for inference. Reported results indicate that CNN and CRNN showed better performance over RNN in accurately detecting and

Table D.1: Summary of the latest works on machine learning-based acoustic detection of UAVs

Reference	Method	Results	Limitations
[30]	MFCC coefficients are fed to RNN, GMM, and CNN.	RNN showed best F-score with 80% followed by GMM with 68% and CNN with 58%.	• Augmented environmental noise. • Multi-UAV detection is absent.
[26]	Preprocessing using ZCR, MFCC, spectral centroid, etc. Extracted features are fed to multi-class SVM.	UAV detection accuracy - 96%	• Single UAV case. • Lacks real-world experiments and background noise.
[27]	Preprocesing using FFT. Trained using PIL and KNN.	PIL - 83% and KNN - 61% accuracy	• Sound clips from 2 UAVs recorded separately.
[28]	PSD using FFT followed by PCA for dimension reduction. Output fed to SVM.	Best accuracy when SIR was greater than 10 dB.	• SVM is more sensitive to bit error rate.
[31]	Normalized STFT features with CNN.	UAV detection accuracy of 98.97%	• Considered only single UAV scenario. • AWGN is added to simulate a noisy environment.
[25]	Preprocessing with MFCC and STFT. Obtained features fed to SVM and CNN.	eSTFT-SVM reported best performance.	• Considered only single UAV case. • Model accuracy is low.
[34]	Audio spectrograms with CNN, RNN, CRNN.	CNN reported best detection accuracy with 96.38% followed by CRNN with 94.72%. Experimented with two different types of UAVs.	• Lacks real-world experiment scenarios. • Doesn't estimate the number of UAVs.
[32]	Audio spectrograms with CNN. Used different RPAS classes individually for measurements.	Greater than 90% detection accuracy.	• Multiple RPAS scenario is absent.
[22]	MFCC and LPCC for feature extraction. Features are fed to SVM with linear, cubic, and quadratic kernels.	MFCC with SVM cubic kernel achieves 96.7% detection accuracy.	• Considers only single UAV scenario.
[24]	Trains neural network on cepstrum coefficients.	Relatively high UAV detection rate.	• Multiple UAV scenarios are absent.
[33]	Mel-spectrograms for feature extraction followed by CNN and CRNNs.	CNN (94.7% accuracy) outperformed CRNN (94.1% accuracy). Experimented with real-world scenarios.	• Multiple UAV scenarios are absent.
[29]	BSS using ICA, PCA, or VMD features. Obtained features are fed to SVM, KNN, and Decision trees	SVM and KNN reported more than 90% accuracy	• Lacks real-world scenarios with background noise.

classifying UAVs. Our work also revolves around a similar approach in which we utilize audio spectrograms to perform multiple UAV detection.

Table D.1 summarizes the latest works in the literature related to machine learning-based acoustic detection of UAVs. As seen from Table D.1, the majority of the literature focuses on detection for a single UAV scenario [25], [26], [30], [31], [32], [22], [33]. The results obtained for detecting a single UAV can widely vary in a multiple UAV scenario. Similarly, the scenarios considered in the literature more or less replicate controlled and well-defined UAV trajectories [26], [34], [30], [31], [29]. Such scenarios may not completely provide a realistic UAV flight trajectory and may affect detection accuracy. Furthermore, some of the techniques described require the use of high-end sophisticated computing infrastructure which may not be always feasible and available [28]. Our work differs from the previous works in detecting multiple UAVs rather than a single UAV. To the best of our knowledge, this is the first time acoustic signatures have been employed to estimate the detection of maximum 10 UAVs in a scene. The scenarios considered comprise multiple UAVs maneuvering in random directions and speeds. Our work also uses the inherent background noise while performing detection. We have included one outdoor measurement that includes background noise from wind and birds chirping. We use supervised learning techniques in this work due to their superior performance in the detection and classification of targets. Although unsupervised techniques have the advantage of extracting the inherent features from unlabeled data, it fails in performance when the requirement calls for the ability to identify specific classes of targets. Further, our custom CNN architecture outperforms the current state-of-the-art machine learning models in terms of accuracy and model size. Owing to the relatively less model size, the custom CNN architecture consumes fewer resources thereby enabling it to be deployed on lightweight edge-computing devices such as Raspberry Pi 4, NVIDIA Jetson Nano, etc. We test the model on these devices and also provide inference time for the same. As such the major contributions of this paper are as follows:

- An UAV acoustic-based dataset is created by utilizing a total of 10 UAVs. An arbitrary number of UAVs are flown randomly within the measurement area and the acoustic field signatures are captured using a cardioid unidirectional microphone.

- Time-frequency algorithms such as continuous wavelet transform (CWT) are applied to transform the recorded acoustic field signatures to spectrogram images.

- A custom lightweight CNN architecture is designed to estimate the number of UAVs in the scene. The performance of the proposed model is compared with state-of-the-art benchmark machine learning models in terms of accuracy and model size.

The remaining sections of this paper are organized as follows: Section D.3 provides the methodology of the proposed approach. Section D.4 describes the measurement setup and details regarding dataset creation. Section D.5 focuses on the data preprocessing and the machine learning algorithm that is used. Section D.6

summarizes the standard benchmark machine learning models that are used to compare the performance of the proposed model. Section D.7 provides an overview of the edge computing devices on which the proposed model is executed. Section D.8 summarizes the results obtained with the proposed approach. Section D.9 discusses the implications of the proposed approach along with limitations and future work. Finally, the paper is concluded in Section D.10.

D.3 Methodology

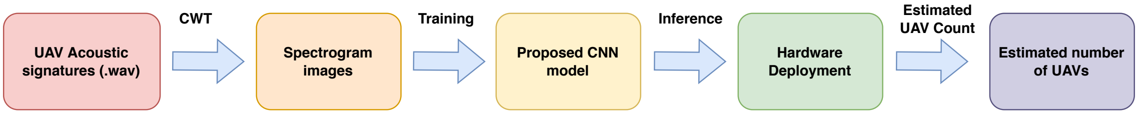


Figure D.1: Block diagram of the proposed work.

The proposed work utilizes the acoustic field generated from UAV rotors and propellers. The high-speed motion of rotors and propellers produces pressure differences leading to the generation of an acoustic field. In the proposed work, these acoustic field signatures are captured with the help of a cardioid microphone. Acoustic signatures from 10 UAV combinations are captured and processed to generate audio spectrogram images. These images are then used to train CNN models to estimate the number of UAVs in the scene. A simplified flow diagram of the methodology is shown in Fig. D.1.

D.4 Measurement setup and dataset

D.4.1 Measurement setup

The primary measurement area is an indoor lab environment that covers a semicircular area of 5 meters in radius. Experiments were designed to capture acoustic signatures from a total of 10 UAV combinations that are flown in a random fashion within the prescribed area. The UAV models employed for the experiment include the DJI Mavic 2 Enterprise [35], DJI Mini 2 [36], DJI Mini SE [37], DJI Mini 3 Pro [38], DJI Tello EDU [39], and SYMA X30 [40]. Except for the DJI Mavic 2 Enterprise, all the other UAVs fall into the 250 grams category and have relatively smaller dimensions (approximately 251 × 362 × 70 mm [38]). The small size of the UAVs makes them an excellent choice for testing the estimation performance in a multi-UAV scenario. The DJI Tello UAVs were operated programmatically to follow a prescribed trajectory. The remaining UAVs were operated manually to fly in a random fashion to simulate a near real-time scenario. Further details regarding the experiment are provided in Table D.2.

The UAV acoustic signatures are recorded using the Shure MV7 microphone [41]. The microphone has a unidirectional (cardioid) type polar pattern with an output impedance of 314 ohms. It has a frequency response ranging from 50 Hz to 16000

Table D.2: Measurement setup and experiment details

Experiment Parameters	**Details**
Semicircular area	Radius: 5 meters
Measurement duration	5 minutes
Microphone	**Shure MV7**
Frequency response	50 Hz - 16000 Hz
Output impedance	314 ohms
Sampling rate	48000 Hz
Total UAVs	**10**
DJI Mavic 2 Enterprise	1
DJI Mini 2	1
DJI Mini SE	2
DJI Mini 3 Pro	1
DJI Telllo EDU	4
SYMA X30	1

Hz with an adjustable gain spanning between 0 and +36 dB [42]. Additional details regarding the microphone are provided in Table D.2. The microphone is mounted on a tripod stand and faced toward the measurement area. As the microphone has a unidirectional cardioid polar pattern, the acoustic disturbances originating from UAVs flying in front of the microphone are captured and amplified. However, the disturbances that originate from the rear end of the microphone are attenuated and hence do not contribute to the output signal. Fig. D.2 shows the microphone setup that was used to capture the acoustic signatures from UAVs.

Each measurement of the experiment consisted of flying an arbitrary number of UAVs in the prescribed area for a duration of 5 minutes. For example, the fifth measurement captured acoustic field signatures from 5 randomly flown UAVs. The sixth measurement involved flying 6 UAVs in a random manner. To improve variability, each measurement of the experiment has been designed to use different types of UAVs as much as possible. However, due to availability constraints we resorted to similar UAV models for measurements that required more number of UAVs. Fig. D.3 depicts a 5 UAV measurement case. To provide additional variability in the acoustic field measurements, we performed the 2nd measurement outdoors. The outdoor measurement area is roughly the same semicircular area of radius 5 meters with additional noise related to wind, birds chirping, etc. Further, the first measurement has been taken independently using 3 different UAV models to increase the number of samples.

D.4.2 Dataset details

The acoustic signatures that are recorded and captured from the experiment are used to create a dataset comprising UAV audio clips. Each recorded acoustic signature is of 5 minute duration. The recorded signatures are carefully trimmed to retain only the portion pertaining to UAV audio. Each trimmed audio signal is of 4 minutes and 45 seconds (285 seconds) with a sample rate of 48000 Hz. To reduce latency and

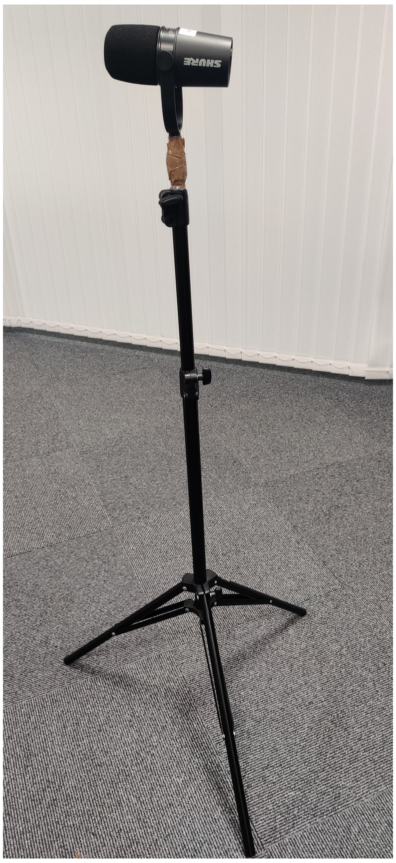


Figure D.2: Shure MV7 setup used for measuring the UAV acoustic signals

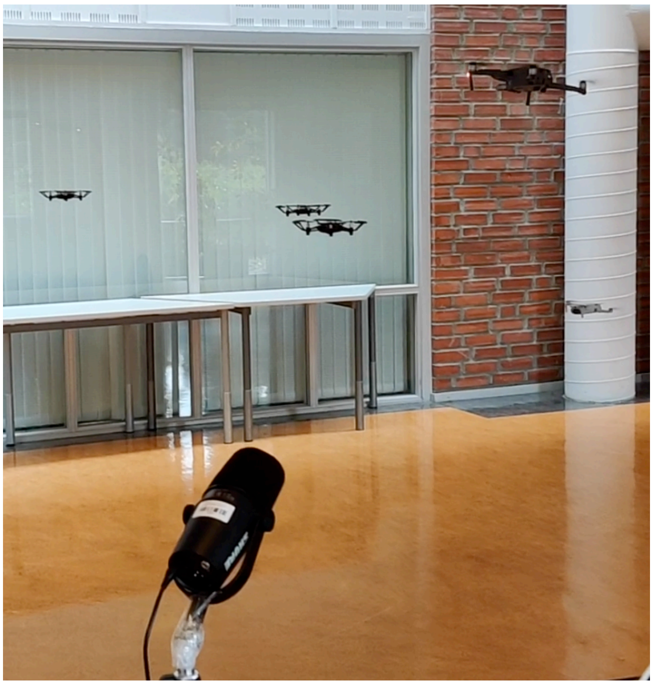


Figure D.3: Capturing acoustic signals for a 5 UAV scenario.

ensure smooth processing, we divide each trimmed signal (285 second duration) into 95 equal parts. Each one of the 95 parts is a 3 second audio clip with a sampling rate of 48000 Hz. In total, the dataset contains 1140 UAV audio clips of 3 second duration.

D.5 Preprocessing and algorithm details

In the preprocessing stage of the proposed approach, various signal processing transformations are applied on the prepared UAV audio dataset. After preprocessing, the resulting spectrogram images are then fed to lightweight CNN models to estimate the number of UAVs present in the scene.

D.5.1 Continuous Wavelet Transform (CWT)

The CWT is a wavelet transform that decomposes a signal into its time and frequency components [43]. Just like the STFT [44], the CWT measures the correlation between the original signal $f(t)$ and the analyzing wavelet ψ. Depending upon the correlation with the original signal, the analyzing wavelet is scaled and dilated by parameters p and q respectively. Assuming the scaling parameter $p > 0$, and dilation parameter q, then the CWT for a signal $f(t)$ is computed as,

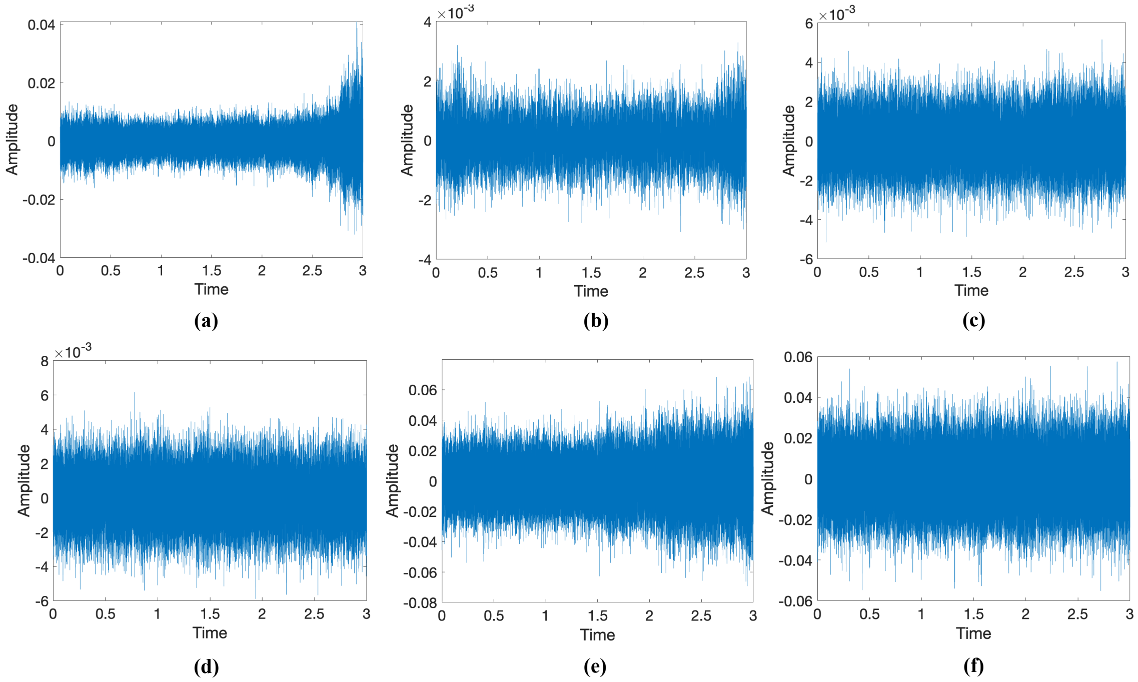
$$C(p, q; f(t), \psi(t)) = \int_{-\infty}^{\infty} f(t)\frac{1}{p}\psi^{*}\left(\frac{t-q}{p}\right) dt, \tag{D.1}$$

where the * represents the complex conjugate [45]. If the CWT is applied to a real signal, then the obtained output is also real-valued. By varying the parameters p and q continuously, we obtain the $C(p, q)$ coefficients which are subsequently used to plot the spectrogram of the signal. In the proposed method, CWT is applied over the trimmed audio clips. As each audio clip is 3 seconds long with a sample rate of 48000 Hz, the resulting spectrogram exhibits time and frequency components corresponding to these parameters. The acoustic signatures along with their corresponding spectrogram outputs are plotted in Fig. D.4 and D.5.

D.5.2 Convolutional Neural Network (CNN)

CNNs are unique deep-learning architectures that utilize artificial neural networks to detect and classify objects from images. In the proposed approach, we develop a custom CNN architecture to extract feature information from spectrogram images.

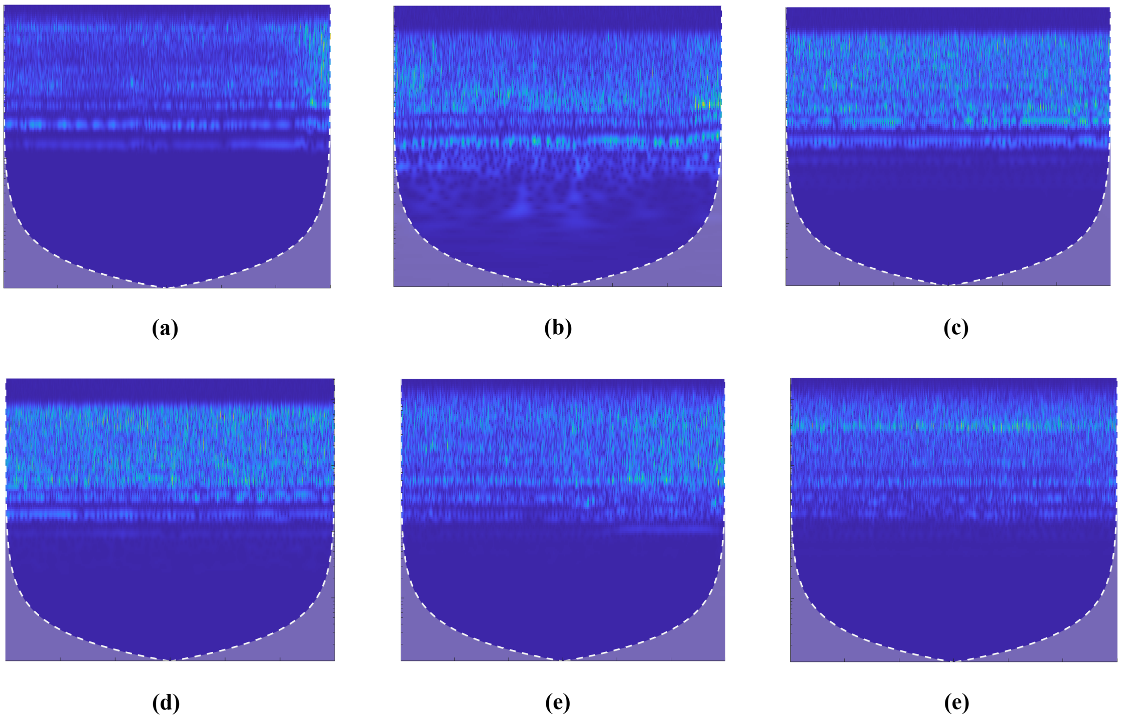
While designing the custom CNN architecture, we first checked the performance by varying the number of layers as 5, 10, 15, and 20 layers. We used 10-fold cross-validation with a data set split of (80, 10, 10) for training, testing, and validation. It was observed that CNNs with 20 layers or more provided better performance as compared to the ones with a lower number of layers. Subsequently, we varied the number of layers along with the image resolution to obtain the best-performing architectures. Table D.3 provides the performance of the CNN architectures with 18 layers and more. The change in performance is also noted with respect to the change in image resolution. It can be observed from Table D.3 that the performance



(a) (b) (c)

(d) (e) (f)

Figure D.4: UAV acoustic signatures for scenarios with (a) 1 UAV (b) 2 UAVs (c) 4 UAVs (d) 6 UAVs (e) 8 UAVs (f) 10 UAVs.



(a) (b) (c)

(d) (e) (e)

Figure D.5: Spectrogram images of scenarios with (a) 1 UAV (b) 2 UAVs (c) 4 UAVs (d) 6 UAVs (e) 8 UAVs (f) 10 UAVs.

gradually increases from 18 layers and peaks around 22 and 24 layers and then gradually decreases. Out of the five CNN architectures (shown in bold in Table D.3) that exhibited greater than 94% average test accuracy, we chose the CNN

Table D.3: Comparison of CNN architectures based on number of layers and image resolution

Sl. No.	No of Layers	Image resolution	Avg. Test Accuracy (%)
1	18	300 × 200 × 3	92.29
2	18	400 × 300 × 3	92.40
3	18	500 × 400 × 3	86.05
4	18	600 × 500 × 3	**94.73**
5	18	700 × 600 × 3	93.55
6	20	300 × 200 × 3	89.37
7	20	400 × 300 × 3	88.50
8	20	500 × 400 × 3	89.56
9	20	600 × 500 × 3	88.89
10	20	700 × 600 × 3	88.00
11	22	300 × 200 × 3	92.19
12	22	400 × 300 × 3	**94.24**
13	22	500 × 400 × 3	**94.74**
14	22	600 × 500 × 3	93.86
15	22	700 × 600 × 3	**94.05**
16	24	300 × 200 × 3	92.98
17	24	400 × 300 × 3	92.98
18	24	500 × 400 × 3	93.56
19	24	600 × 500 × 3	93.86
20	24	700 × 600 × 3	**94.24**
21	26	300 × 200 × 3	91.13
22	26	400 × 300 × 3	91.61
23	26	500 × 400 × 3	89.76
24	26	600 × 500 × 3	86.34
25	26	700 × 600 × 3	88.11

architecture (22 layers, 500 × 400 × 3) for further analysis after considering other performance metrics and parameters.

The proposed CNN architecture that is made of 22 layers is shown in Fig. D.6 and Table D.4. The spectrogram image obtained after performing CWT on the audio clips have a resolution of 836 × 716 × 3 pixels. To be trained by the custom CNN architecture, these spectrogram images are resized to 500×400×3 pixels. The resized images are then fed to the proposed CNN architecture through the input layer. The proposed architecture consists of convolutional layers with kernel dimensions 4 × 4. We primarily use 8 or 16 convolutional kernels to extract the feature embeddings. The final convolutional layer however additionally uses dilation by a factor of 2. The process of dilation intentionally expands the kernel size by introducing holes between adjacent elements as shown in Fig. D.7. This provides a larger field of view that in turn helps in capturing intrinsical sequence information [46].

To expand the network with additional layers without compromising on performance, our model employs residual blocks. Our model uses a total of 2 residual layers. The architecture of the residual blocks used in the proposed architecture

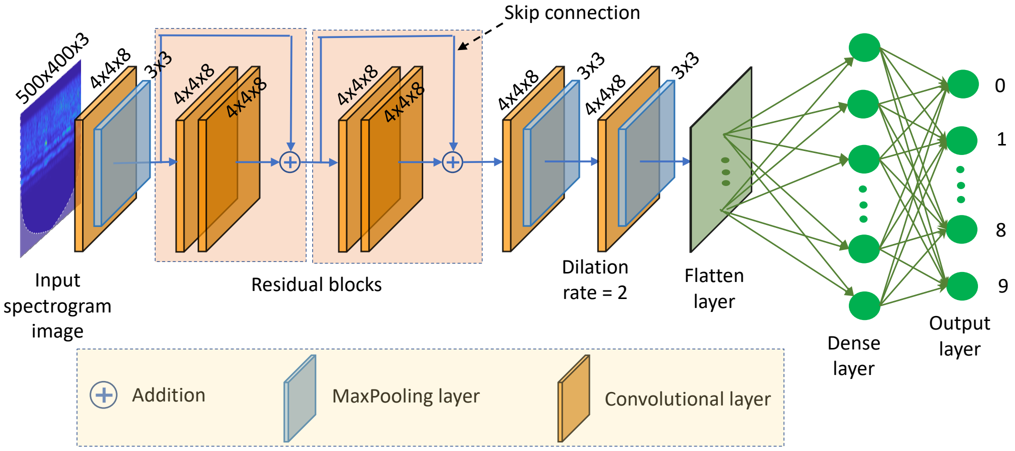
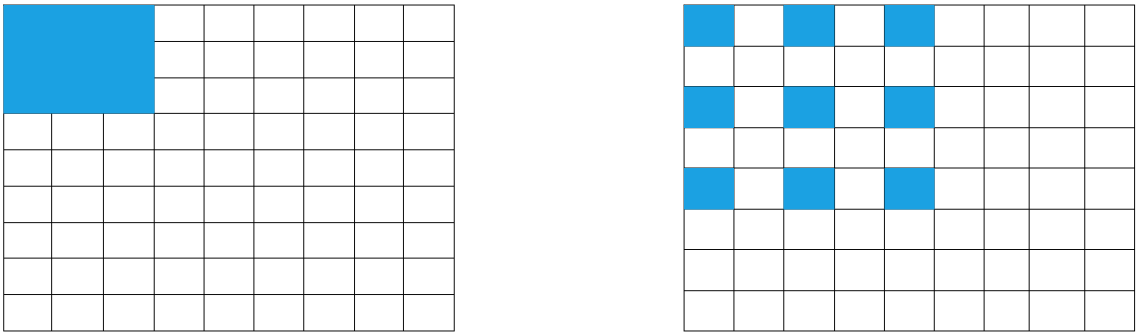


Figure D.6: Proposed CNN architecture.

Table D.4: Layerwise architecture details of the proposed CNN model

No.	Layer	Output Size	Parameter
1	Input	[(None, 500, 400, 3)]	0
2	Batch Normalization (1)	[(None, 500, 400, 3)]	12
3	Conv2D (1)	[(None, 500, 400, 8)]	392
4	MaxPooling2D (1)	[(None, 166, 133, 8)]	0
5	Conv2D (2)	[(None, 166, 133, 8)]	1032
6	Batch Normalization (2)	[(None, 166, 133, 8)]	32
7	Conv2D (3)	[(None, 166, 133, 8)]	1032
8	Batch Normalization (3)	[(None, 166, 133, 8)]	32
9	Add (1)	[(None, 166, 133, 8)]	0
10	Activation (1)	[(None, 166, 133, 8)]	0
11	Conv2D (4)	[(None, 166, 133, 16)]	1032
12	Batch Normalization (4)	[(None, 166, 133, 8)]	32
13	Conv2D (5)	[(None, 166, 133, 16)]	1032
14	Batch Normalization (5)	[(None, 166, 133, 8)]	32
15	Add (2)	[(None, 166, 133, 8)]	0
16	Activation (2)	[(None, 166, 133, 8)]	0
17	Conv2D (6)	[(None, 166, 133, 16)]	2064
18	MaxPooling2D (2)	[(None, 55, 44, 16)]	0
19	Conv2D (7), Dilation rate = 2	[(None, 55, 44, 64)]	16448
20	MaxPooling2D (3)	[(None, 18, 14, 64)]	0
21	Flatten	[(None, 16128)]	0
22	Dense	[(None, 10)]	161290

is depicted in Fig. D.6. Additionally, 3 × 3 max pooling is used throughout the architecture. Max pooling downsamples the input feature representation [47]. It essentially removes translational invariances from the input representation thereby improving computational efficiency for further layers. The proposed model also utilizes batch normalization at the input layer and residual layers as observed from Table D.4. Batch normalization resolves the problem of internal covariate shift [48] by standardization of the input distribution that involves re-centering and re-scaling. The final layers comprise the flatten layer and the dense layer. The flatten



(a) Dilation factor = 1

(b) Dilation factor = 2

Figure D.7: Effect of dilated convolution for a 3x3 kernel on a 9x9 feature map.

layer transforms the input vector to a 1-dimensional output which is subsequently fed into the dense layer. The dense layer outputs class probabilities which are finally used for detection and classification tasks. In the proposed architecture, since we are estimating a maximum number of 10 UAVs, there are 10 outputs from the dense layer.

The proposed CNN model is trained using the Adam optimizer [49] with the tanh activation function. We have adjusted the learning rate to 0.001 and batch size to 16 to reduce fluctuations in the accuracy/loss curve during training. Training is performed using the Keras deep learning library [50] on two Tesla V100-SXM3 GPU with 32 GB RAM [51].

D.6 Benchmark Models

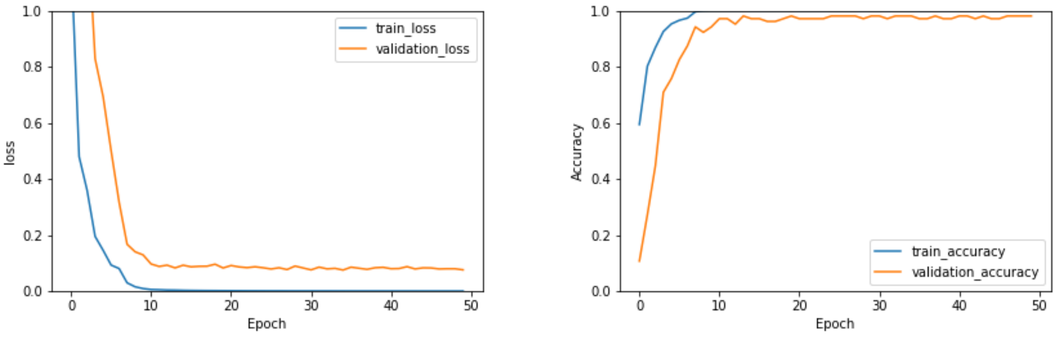
Benchmark models are state-of-the-art models that have distinct architectural features. For example, DenseNets [52] are special CNNs where the feature maps from each layer are fed to all the subsequent layers thereby preserving the feed-forward nature of the network. ResNet architecture [53] introduces residual blocks to improve performance. The residual blocks are made up of skip connections that retain the abstractions lost in the standard path. The efficientNet family of CNNs utilizes uniform scaling of the depth, width, and resolution of the network to achieve better accuracy. MobileNet [54] models on the other hand are optimized to provide faster operations on mobile and embedded devices. These models have a low memory footprint and offer a better tradeoff between resource utilization and accuracy. We assess the performance of the proposed model by comparing it with these existing benchmark models. We compare it with 23 benchmark models that include models from DenseNet [52], EfficientNet [55], Inception [56], MobileNet [54], ResNet [53], NASNetMobile [57], VGG [58], and Xception [59]. The benchmark models are pre-trained on the ImageNet dataset [60]. The input image resolution fed to the benchmark models has a resolution of 224 × 224 × 3. We compare the proposed model with benchmark models in terms of total parameters, model size, average test accuracy, and number of floating point operations per second (FLOPs). The benchmark models are also further deployed on edge computing devices to measure inference time.

D.7 Hardware Deployment

To obtain real-time performance with edge computing devices, we deploy the proposed model on three embedded devices, namely, the Raspberry Pi 4 Model B, the NVIDIA Jetson AGX Xavier, and the NVIDIA Jeton Nano. The Raspberry Pi 4 Model B board contains a quad-core ARM Cortex-A72 processor with 1/2/4/8 GB of RAM. The board is well equipped with various communication interfaces such as Bluetooth 5.0, BLE, and 2.4/5.0 GHz wireless LAN for wireless information transfer. Additionally, the board also provides two USB 3.0, two USB 2.0, and a Gigabit Ethernet port to ensure seamless interfacing with other devices [61]. As compared to Raspberry Pi 4, the NVIDIA Jetson Nano comes with a 128-core Maxwell GPU architecture and quad-core ARM Cortex A5 CPU. With 4 GB RAM and support for multiple interfaces such as USB 2.0 Micro-B, USB 3.0, Gigabit Ethernet, I2C, I2S, SPI, and UART, the Jetson Nano serves to be an excellent choice for high computing edge computing devices [62]. For edge applications that require even more computing capability, the NVIDIA Jetson AGX Xavier is preferred. The Jetson AGX Xavier houses a 512-core Volta GPU architecture and an 8-core Carmel ARM CPU along with 32 GB RAM. It has dedicated deep learning and vision accelerators for various machine learning and computer vision tasks. To interface with other peripheral devices, the Jetson AGX Xavier provides standards such as USB-C, USB 2.0, UART, and RJ45 [63]. To obtain the inference time on the various edge computing devices, the proposed model and benchmark models are first converted to their equivalent Tensorflow Lite versions. TensorFlow Lite [64] is an open-source software developed by Tensorflow to deploy pre-trained models on edge computing devices. By converting the model to its equivalent TensorFlow Lite format, the model is optimized for inference time and model size allowing seamless deployment on various embedded devices. After converting to TensorFlow Lite versions, the proposed model and benchmark models are deployed on these embedded devices to obtain the inference time. The time taken to predict the exact number of UAVs from the spectrogram images is collected and the average inference time is calculated.

D.8 Results

The spectrogram images obtained after applying the CWT transform are used to train the proposed CNN model. We utilize 80% of the dataset for training, 10% for validation, and the remaining 10% for testing. We used 10-fold cross-validation [65] where each fold is trained for 50 epochs with a batch size of 16 and a learning rate of 0.001. The 10-fold cross-validation utilizes 80% of the dataset for training and 10% for validation. The testing is performed on the remaining 10% of the dataset to obtain detection accuracy. Fig. D.8a and D.8b show the loss and accuracy curves

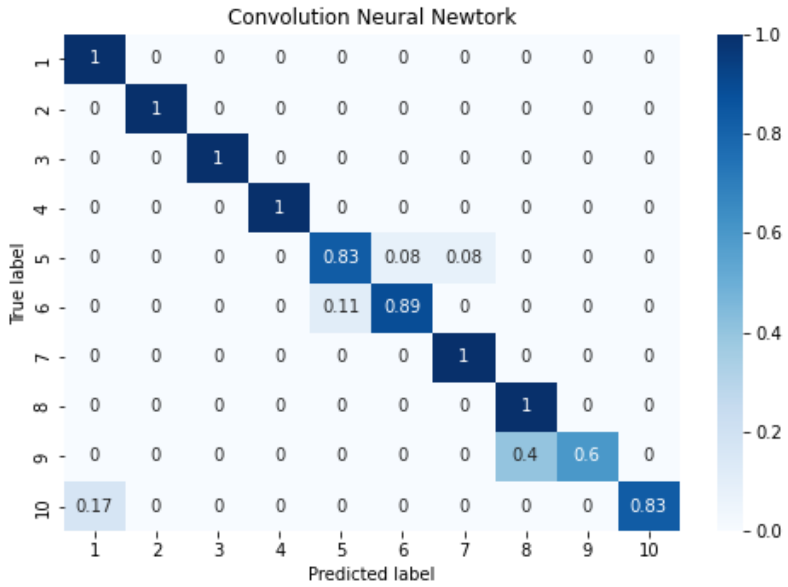


(a) Loss curve

(b) Accuracy curve

Figure D.8: Loss and accuracy curves.

obtained during training. It can be observed from Fig. D.8a that the training and validation loss decrease significantly after 10 epochs. Correspondingly, the accuracy curves for training and validation converge close to 1 after 10 epochs indicating that the proposed model requires less training time.



	1	2	3	4	5	6	7	8	9	10
1	1	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0
5	0	0	0	0	0.83	0.08	0.08	0	0	0
6	0	0	0	0	0.11	0.89	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0
8	0	0	0	0	0	0	0	1	0	0
9	0	0	0	0	0	0	0	0.4	0.6	0
10	0.17	0	0	0	0	0	0	0	0	0.83

Figure D.9: Confusion matrix for the proposed model.

Upon training, the model performance was evaluated on the test set. Fig. D.9 depicts the confusion matrix obtained after evaluating the proposed model on the test set. It can be observed that the proposed model is able to correctly predict the number of UAVs for more than 90% of the cases. In the remaining cases, the model incorrectly estimates the number of UAVs present in the scene. This might be due to the superposition of acoustic signatures from similar UAV models that can render the obtained signal unresolvable. The performance of the proposed model is also compared with standard benchmark machine learning models as shown in Table D.5. We used the same data set split while calculating the performance metrics on the benchmark models. The spectrogram images are resized to 224×224×3 pixels before

Table D.5: Performance metrics comparison between proposed CNN model and benchmark models

Sl. No.	Model	Total Parameters	Avg. Test Accuracy (%)	Model Size (MB)	Floating Point Operations (GFLOPs)
1	DenseNet121 [52]	$7,047,754$	83.77	28.98	2.88
2	DenseNet169 [52]	$12,659,530$	87.28	51.76	3.42
3	DenseNet201 [52]	$18,341,194$	88.68	74.68	4.37
4	EfficientNetB0 [55]	$4,062,381$	83.85	16.74	0.4
5	EfficientNetB1 [55]	$6,588,049$	81.05	27.01	0.59
6	EfficientNetB2 [55]	$7,782,659$	80.08	31.80	0.68
7	EfficientNetB3 [55]	$10,798,905$	84.21	43.94	0.99
8	EfficientNetB4 [55]	$17,691,753$	78.24	71.66	1.54
9	EfficientNetB5 [55]	$28,534,017$	79.73	115.20	2.41
10	EfficientNetB6 [55]	$40,983,193$	82.45	165.18	3.43
11	EfficientNetB7 [55]	$64,123,297$	81.40	257.98	5.27
12	InceptionResnetV2 [66]	$54,352,106$	77.63	218.77	6.55
13	InceptionV3 [56]	$21,823,274$	78.42	87.97	2.89
14	MobileNetV2 [54]	$2,270,794$	86.75	9.49	0.32
15	MobileNetV3Large [67]	$4,239,242$	86.92	17.42	0.23
16	MobileNetV3Small [67]	$1,540,218$	81.84	6.55	0.06
17	NASNetMobile [57]	$4,280,286$	77.71	18.48	0.27
18	ResNet101V2 [53]	$42,647,050$	89.64	171.37	8.28
19	ResNet152V2 [53]	$58,352,138$	87.63	234.50	12.5
20	ResNet50V2 [53]	$23,585,290$	90.35	94.82	3.97
21	VGG16 [58]	$14,719,818$	87.98	58.98	15.5
22	VGG19 [58]	$20,029,514$	87.98	80.22	19.6
23	Xception [59]	$20,881,970$	82.89	83.96	0.36
24	**Proposed model**	**184,462**	**93.3**	**2.34**	**0.25**

providing it as input to the standard benchmark models. As observed in Table D.5, the proposed model achieves a relatively high test accuracy of 93.33% as compared to the benchmark models. It can also be observed from Table D.5, that the proposed model requires just 2.34 MB of storage space as compared to the benchmark models ensuring seamless portability and deployability on various edge computing devices. Additionally, the total parameters employed by our model are less as compared to other benchmark models. Moreover, the majority of the total parameters used by the proposed model are trainable parameters showing efficient utilization of parameters. Table D.5 also lists the computational performance of our model with respect to other benchmark models in terms of the number of floating point operations (FLOPs) [68]. The FLOP count is measured as GFLOPs where 1 GFLOP is equal to 10^9 FLOPs. The FLOP count is obtained by using standard open-source software available from PyTorch [69] and TensorFlow [70]. It can be observed that our model has a relatively less number of FLOPs as compared to most of the benchmark models. Specifically, MobileNetV3Small and MobileNetV3Large have lower FLOP counts compared to the proposed model. This reduction in computational cost might be due to the width and resolution multiplier parameter introduced in the MobileNet series [54].

The proposed model has also been deployed on edge computing devices such as

Raspberry Pi 4 Model B, NVIDIA Jetson Nano, and NVIDIA Jetson AGX Xavier. We perform inference time calculation of the proposed model on all these devices. The inference time calculation can serve as a useful reference when deciding the deployment feasibility of the proposed model for time-critical applications. In all three devices, we used a randomly selected test set which is 10% of the overall dataset. The trained model was executed on the Raspberry Pi 4 board to obtain an average inference time of about 127 milliseconds over 10 iterations. On the other hand, the execution on the Jetson Nano reported an average inference time of about 219 milliseconds. This is expected as the TensorFlow Lite models are not utilizing the GPU resources onboard the Jetson Nano. TensorFlow Lite does not support CUDA for GPU operations [71]. However, the Jetson devices only support CUDA for GPU operations [72] and hence the observed increase in inference time is expected. Similarly, the trained model was able to provide an average inference time of about 81.4 milliseconds, when executed on the NVIDIA Jetson AGX Xavier. The Jetson AGX Xavier showcased faster inference time as compared to the Raspberry Pi 4 and Jetson Nano. Even though the GPU is not utilized, the faster inference time can be attributed to the availability of increased RAM of about 32 GB which can increase the performance of the system. Further, the Jetson AGX Xavier also has access to additional computing resources as compared to the other two devices. Table D.6 lists the inference time obtained for the proposed model along with the benchmark models on various edge computing devices. We used the same data set split while calculating the inference time on the benchmark models. It can be observed that the proposed model is faster than most of the benchmark models on all three embedded devices. The MobileNet series of models however have a lower inference time than the proposed model. This might be due to the width and resolution multiplier parameter in the MobileNet series that reduces the computational cost of the model.

D.9 Discussion

In this work, we have provided a robust solution to estimate the number of UAVs in a scene. The current setup employs only one unidirectional cardioid-type microphone to estimate the number of UAVs. It is to be noted that since the polar pattern of the microphone follows a cardioid pattern, the acoustic disturbances originating from UAVs flying at the rear of the microphone are attenuated. This can severely impact the estimated UAV number. A more practical approach to overcome this limitation is to position multiple cardioid microphones such that the acoustic disturbances originating from the full 360° of the scene are captured. Employing microphones that exhibit an omnidirectional polar pattern can also be utilized so that acoustic disturbances from all directions are captured without significant signal attenuation.

It can be observed from Table D.5 and Table D.6, that the proposed CNN architecture provides relatively high accuracy and fast inference time on embedded hardware all the while consuming fewer resources. The proposed model can thus be employed for time-critical and resource-constrained UAV detection scenarios. High

Table D.6: Inference time calculation on various edge computing devices

Sl. No	**Model**	**Inference time (seconds)**		
		Raspberry Pi	**Jetson Nano**	**Jetson AGX Xavier**
1	DenseNet121 [52]	0.692	0.841	0.413
2	DenseNet169 [52]	0.832	1.009	0.483
3	DenseNet201 [52]	1.088	1.275	0.614
4	EfficientNetB0 [55]	0.389	0.372	0.116
5	EfficientNetB1 [55]	0.591	0.546	0.178
6	EfficientNetB2 [55]	0.624	0.592	0.180
7	EfficientNetB3 [55]	0.843	0.794	0.248
8	EfficientNetB4 [55]	1.162	1.147	0.359
9	EfficientNetB5 [55]	1.724	1.672	0.528
10	EfficientNetB6 [55]	2.339	2.173	0.832
11	EfficientNetB7 [55]	3.252	2.971	1.158
12	InceptionResNetV2 [66]	1.637	1.828	0.797
13	InceptionV3 [56]	0.700	0.819	0.378
14	MobileNetV2 [54]	0.088	0.112	0.046
15	MobileNetV3Large [67]	0.072	0.090	0.036
16	MobileNetV3Small [67]	0.022	0.028	0.011
17	NASNetMobile [57]	0.320	0.256	0.125
18	ResNet50V2 [53]	0.912	1.008	0.440
19	ResNet101V2 [53]	1.879	2.053	0.858
20	ResNet152V2 [53]	2.820	3.159	1.285
21	VGG16 [58]	3.903	4.042	1.699
22	VGG19 [58]	5.031	6.981	2.184
23	Xception [59]	1.236	2.206	0.520
24	**Proposed model**	**0.127**	**0.219**	**0.081**

detection performance coupled with real-time scenarios also suggests that the proposed technique can be deployed in practical ground control stations to function as an anti-UAV detection system. It can be inferred from the obtained results that the proposed technique is capable of detecting more than 10 UAVs in a dynamic real-time scenario given additional UAV information. In the future, the accuracy of the proposed technique can be improved by utilizing other sensor modalities. With the help of additional sensors, the work can also be extended to identify the UAV model and/or type.

D.10 Conclusion

In this article, we addressed the problem of accurately estimating the total number of UAVs present in a scene. We developed a UAV acoustic dataset to recreate a real-world scenario comprising of 10 UAV combinations flown in a random manner. The acoustic information from the dataset was preprocessed using time-frequency transformations to obtain their respective spectrogram images. The generated spectrogram images are then fed into a custom lightweight CNN model to estimate the number of UAVs in the scene. The proposed model provides a high average test accuracy in accurately estimating the number of UAVs. Subsequently, the proposed model has also been executed on various edge computing devices to measure inference time performance. In the future, this work can be extended to identify the UAV model and/or type by utilizing information from additional sensors.

References

[1] Praveen Kumar Reddy Maddikunta et al. Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges. *IEEE Sensors Journal*, 21(16):17608–17619, 2021.

[2] Chunxue Wu, Bobo Ju, Yan Wu, Xiao Lin, Naixue Xiong, Guangquan Xu, Hongyan Li, and Xuefeng Liang. Uav autonomous target search based on deep reinforcement learning in complex disaster scene. *IEEE Access*, 7:117227–117245, 2019.

[3] Junbiao Zhang et al. Trajectory prediction of hypersonic glide vehicle based on empirical wavelet transform and attention convolutional long short-term memory network. *IEEE Sensors Journal*, 22(5):4601–4615, 2022.

[4] Rodrigo Saar de Moraes and Edison Pignaton de Freitas. Multi-uav based crowd monitoring system. *IEEE Transactions on Aerospace and Electronic Systems*, 56(2):1332–1345, 2020.

[5] A. N. Wilson, Abhinav Kumar, Ajit Jha, and Linga Reddy Cenkeramaddi. Embedded sensors, communication technologies, computing platforms and machine learning for UAVs: A review. *IEEE Sensors Journal*, 22(3):1807–1826, 2022.

[6] Jian Wang, Yongxin Liu, and Houbing Song. Counter-unmanned aircraft system(s) (c-uas): State of the art, challenges, and future trends. *IEEE Aerospace and Electronic Systems Magazine*, 36(3):4–29, 2021.

[7] Wei Nie et al. UAV detection and identification based on WiFi signal and RF fingerprint. *IEEE Sensors Journal*, 21(12):13540–13550, 2021.

[8] Tianyuan Yang et al. An adaptive radar signal processor for UAVs detection with super-resolution capabilities. *IEEE Sensors Journal*, 21(18):20778–20787, 2021.

[9] Alexander Sedunov et al. Stevens drone detection acoustic system and experiments in acoustics UAV tracking. In *Proc. IEEE International Symposium on Technologies for Homeland Security (HST)*, pages 1–7, 2019.

[10] Jesse Callanan, Payam Ghassemi, James DiMartino, Maulikkumar Dhameliya, Christina Stocking, Mostafa Nouh, and Souma Chowdhury. Ergonomic impact of multi-rotor unmanned aerial vehicle noise in warehouse environments. *Journal of Intelligent & Robotic Systems*, 100:1309–1323, 2020.

[11] Jesse Callanan, Rayhaan Iqbal, Revant Adlakha, Amir Behjat, Souma Chowdhury, and Mostafa Nouh. Large-aperture experimental characterization of the acoustic field generated by a hovering unmanned aerial vehicle. *The Journal of the Acoustical Society of America*, 150(3):2046–2057, 2021.

[12] Rayhaan Iqbal, Amir Behjat, Revant Adlakha, Jesse Callanan, Mostafa Nouh, and Souma Chowdhury. Efficient training of transfer mapping in physics-infused machine learning models of uav acoustic field. In *AIAA SCITECH 2022 Forum*, page 0384, 2022.

[13] Jianfei Tong, Wei Xie, Yu-Hen Hu, Ming Bao, Xiaodong Li, and Wei He. Estimation of low-altitude moving target trajectory using single acoustic array. *The Journal of the Acoustical Society of America*, 139(4):1848–1858, 2016.

[14] Alexander Sedunov, Alexander Sutin, and Hady Salloum. Application of cross-correlation methods for passive acoustic unmannded aierial vehicle detection and tracking. *The Journal of the Acoustical Society of America*, 140(4):3119–3119, 2016.

[15] Vladimir Kartashov et al. Use of acoustic signature for detection, recognition and direction finding of small unmanned aerial vehicles. In *Proc. IEEE International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, pages 1–4, 2020.

[16] Torea Blanchard, J-H Thomas, and Kosai Raoof. Acoustic localization and tracking of a multi-rotor unmanned aerial vehicle using an array with few microphones. *The Journal of the Acoustical Society of America*, 148(3):1456–1467, 2020.

[17] Giuseppe Ciaburro, Gino Iannace, and Amelia Trematerra. Research for the presence of unmanned aerial vehicle inside closed environments with acoustic measurements. *Buildings*, 10(5), 2020.

[18] Jian Fang, Anthony Finn, Ron Wyber, and Russell SA Brinkworth. Acoustic detection of unmanned aerial vehicles using biologically inspired vision processing. *The Journal of the Acoustical Society of America*, 151(2):968–981, 2022.

[19] Michael J Bianco, Peter Gerstoft, James Traer, Emma Ozanich, Marie A Roch, Sharon Gannot, and Charles-Alban Deledalle. Machine learning in acoustics: Theory and applications. *The Journal of the Acoustical Society of America*, 146(5):3590–3628, 2019.

[20] Aro Ramamonjy, Eric Bavu, Alexandre Garcia, and Sébastien Hengy. A distributed network of compact microphone arrays for drone detection and tracking. *The Journal of the Acoustical Society of America*, 141(5):3651–3651, 2017.

[21] Bilal Taha and Abdulhadi Shoufan. Machine learning-based drone detection and classification: State-of-the-art in research. *IEEE Access*, 7:138669–138682, 2019.

[22] Muhammad Zohaib Anwar, Zeeshan Kaleem, and Abbas Jamalipour. Machine learning inspired sound-based amateur drone detection for public safety applications. *IEEE Transactions on Vehicular Technology*, 68(3):2526–2534, 2019.

[23] Michael W Berry, Azlinah Mohamed, and Bee Wah Yap. *Supervised and unsupervised learning for data science.* Springer, 2019.

[24] Valentin V Gravirov, Ruslan A Zhostkov, and Dmitriy A Presnov. Acoustic fields of unmanned aerial vehicles in the tasks of passive detection. *The Journal of the Acoustical Society of America*, 149(4):A35–A35, 2021.

[25] Bowon Yang et al. UAV detection system with multiple acoustic nodes using machine learning models. In *Proc. IEEE International Conference on Robotic Computing (IRC)*, pages 493–498, 2019.

[26] Andrea Bernardini, Federica Mangiatordi, Emiliano Pallotti, and Licia Capodiferro. Drone detection by acoustic signature identification. *Electronic Imaging*, 2017(10):60–64, 2017.

[27] Juhyun Kim, Cheonbok Park, Jinwoo Ahn, Youlim Ko, Junghyun Park, and John C. Gallagher. Real-time uav sound detection and analysis system. In *IEEE Sensors Applications Symposium (SAS)*, pages 1–5, 2017.

[28] Xuejun Yue, Yongxin Liu, Jian Wang, Houbing Song, and Huiru Cao. Software defined radio and wireless acoustic networking for amateur drone surveillance. *IEEE Communications Magazine*, 56(4):90–97, 2018.

[29] Wenshuai Wang, Kuangang Fan, Qinghua Ouyang, and Ye Yuan. Acoustic uav detection method based on blind source separation framework. *Applied Acoustics*, 200:109057, 2022.

[30] Sungho Jeon, Jong-Woo Shin, Young-Jun Lee, Woong-Hee Kim, YoungHyoun Kwon, and Hae-Yong Yang. Empirical study of drone sound detection in real-life environment with deep neural networks. In *Proc. European Signal Processing Conference (EUSIPCO)*, pages 1858–1862, 2017.

[31] Yoojeong Seo, Beomhui Jang, and Sungbin Im. Drone detection using convolutional neural networks with acoustic stft features. In *Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2018.

[32] Luke Russell, Rafik Goubran, and Felix Kwamena. Emerging urban challenge: Rpas/uavs in cities. In *Proc. International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 546–553, 2019.

[33] Pietro Casabianca and Yu Zhang. Acoustic-based UAV detection using late fusion of deep neural networks. *Drones*, 5(3), 2021.

[34] Sara Al-Emadi, Abdulla Al-Ali, Amr Mohammad, and Abdulaziz Al-Ali. Audio based drone detection and identification using deep learning. In *Proc. International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 459–464, 2019.

[35] DJI Mavic 2. [Online]. Available: https://www.dji.com/no/mavic-2-enterprise.

[36] DJI Mini 2. [Online]. Available: https://www.dji.com/no/mini-2.

[37] DJI Mini SE. [Online]. Available: https://www.dji.com/no/mini-se.

[38] DJI Mini 3 Pro. [Online]. Available: https://www.dji.com/no/mini-3-pro.

[39] DJI Tello EDU. [Online]. Available: https://store.dji.com/no/product/tello.

[40] SYMA X30 Foldable Drone. [Online]. Available: https://www.symatoys.com/goodshow/x30-syma-x30-foldable-drone.html.

[41] Shure MV7. [Online]. Available: https://www.shure.com/en-MEA/products/microphones/mv7?variant=MV7-K.

[42] Shure MV7 User Guide. [Online]. Available: https://pubs.shure.com/guide/MV7/en-US.

[43] Yoon Young Kim and Eung-Hun Kim. Effectiveness of the continuous wavelet transform in the analysis of some dispersive elastic waves. *The Journal of the Acoustical Society of America*, 110(1):86–94, 2001.

[44] L. Durak and O. Arikan. Short-time fourier transform: two fundamental properties and an optimal implementation. *IEEE Transactions on Signal Processing*, 51(5):1231–1242, 2003.

[45] Continuous Wave Transform (CWT. [Online]. Available: https://www.mathworks.com/help/wavelet/gs/continuous-wavelet-transform-and-scale-based-analysis.html.

[46] Aveen Dayal, Sreenivasa Reddy Yeduri, Balu Harshavardan Koduru, Rahul Kumar Jaiswal, J Soumya, MB Srinivas, Om Jee Pandey, and Linga Reddy Cenkeramaddi. Lightweight deep convolutional neural network for background sound classification in speech signals. *The Journal of the Acoustical Society of America*, 151(4):2773–2786, 2022.

[47] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proc. International Conference on Machine Learning (ICML)*, pages 111–118, 2010.

[48] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proc. International Conference on Machine Learning (ICML)*, pages 448–456. PMLR, 2015.

[49] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[50] Keras. [Online]. Available: https://keras.io.

[51] Tesla V100 GPU. [Online]. https://www.nvidia.com/en-us/data-center/v100/.

[52] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision – ECCV 2016*, pages 630–645, Cham, 2016. Springer International Publishing.

[54] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[55] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proc. International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[56] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[57] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proc. IEEE CVPR*, pages 8697–8710, 2018.

[58] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[59] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proc. IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[60] ImageNet. [Online]. Available: https://www.image-net.org.

[61] Raspberry Pi 4 Model B - Datasheet. [Online]. Available: https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-product-brief.pdf.

[62] NVIDIA Jetson Nano. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit.

[63] NVIDIA Jetson AGX Xavier. [Online]. Available: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-agx-xavier/.

[64] TensorFlow Lite. [Online]. Available: https://www.tensorflow.org/lite.

[65] [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html.

[66] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proc. AAAI conference on artificial intelligence*, volume 31, 2017.

[67] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proc. IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

[68] Hojun Son and James Weiland. Semantic segmentation optimized for low compute embedded devices. *IEEE Access*, 10:96514–96525, 2022.

[69] Torchstat (PyTorch). [Online]. https://pypi.org/project/torchstat/.

[70] Model Profiler (TensorFlow). [Online]. Available: https://pypi.org/project/model-profiler/.

[71] GPU Delegates - TensorFlow Lite. [Online]. Available: https://www.tensorflow.org/lite/performance/delegates.

[72] Shan Ullah and Deok-Hwan Kim. Benchmarking jetson platform for 3d point-cloud and hyper-spectral image classification. In *Proc. IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 477–482, 2020.