Audio Based Drone Detection and Identification using Deep Learning

Sara Al-Emadi, Abdulla Al-Ali, Amr Mohammad, Abdulaziz Al-Ali
Department of Computer Science and Engineering
Qatar University
Doha, Qatar
saraalemadi@ieee.org, [abdulla.alali, amrm, a.alali]@qu.edu.qa

***Abstract*—In recent years, unmanned aerial vehicles (UAVs) have become increasingly accessible to the public due to their high availability with affordable prices while being equipped with better technology. However, this raises a great concern from both the cyber and physical security perspectives since UAVs can be utilized for malicious activities in order to exploit vulnerabilities by spying on private properties, critical areas or to carry dangerous objects such as explosives which makes them a great threat to the society. Drone identification is considered the first step in a multi-procedural process in securing physical infrastructure against this threat. In this paper, we present drone detection and identification methods using deep learning techniques such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and Convolutional Recurrent Neural Network (CRNN). These algorithms will be utilized to exploit the unique acoustic fingerprints of the flying drones in order to detect and identify them. We propose a comparison between the performance of different neural networks based on our dataset which features audio recorded samples of drone activities. The major contribution of our work is to validate the usage of these methodologies of drone detection and identification in real life scenarios and to provide a robust comparison of the performance between different deep neural network algorithms for this application. In addition, we are releasing the dataset of drone audio clips for the research community for further analysis.**

***Index Terms*—Drone, UAVs, Acoustic fingerprinting, Drone Audio Dataset, Artificial Intelligence, Machine Learning**

I. Introduction and Related Work

Due to the development of drone technology, the popularity of drones is rapidly increasing as they are becoming more compact in size, easier to operate and widely available for anyone with a desire to use them. Many utilize these drones for recreational purposes such as photography and cinematography, others are studying their behavior and improving on them in order to integrate them in our day to day applications. On the other hand, it has been observed that drones can be utilized for malicious activities to harm targeted individuals and the public. Recently, an incident was reported to authorities in which an explosive equipped drone was hovering over a great crowd in a formal occasion in Venezuela targeting a high profile personnel and the general public. In this incident the drone dropped a number of attached explosives randomly which consequently injured civilians on scene [1]. In addition to the safety issues associated with drones, they can be used to violate security terms and conditions as it has been witnessed in an incident in which smugglers flew drones with illegal drugs and cell phones over prison facilities [2]. Similarly, privacy concerns arose with the introduction of drones where in multiple incidents they were used to spy and record clips of people in their private properties [3][4][5].

As with any device, unintentional accidents beyond the drone operator's control can occur. For example, the connection between the controller and the drone can be lost causing the drone to drift away from the predefined path and crash into its surrounding causing serious damages. This scenario occurred recently in an incident where a drone have severely injured a toddler after crashing into a children's playground [6].

A key objective of this paper is to introduce an autonomous system that in addition to *detection*, is able to *identify* drones based on their acoustic signatures. The identification of the drones is necessary in order to determine the type of the drone which will aid in the process of distinguishing whether the drone flying within a restricted area is an authorized or an unauthorized drone.

There is now a substantial body of research on the application of drone detection using different technologies such as the RF signals [7] [8], a GSM passive coherent location system [9] and a digital TV based bi-static radar [10]. Furthermore, few researchers focused their studies on drone detection using audio characteristics such as a research carried out by the authors in [11], where they have proposed a methodology using digital signal processing (DSP) to detect the presence of drones in an area. Another research serving the same purpose was conducted in [12], in which authors opted for combining DSP with Machine Learning algorithms such as the Support Vector Machine (SVM) algorithm. The researchers have reported the effectiveness of using SVM in drone detection which have yielded high accuracy, yet, the research was limited to a specific background noises. Furthermore, SVM requires manual extraction and optimization of hand-crafted features to fine tune the algorithm, this is an additional step to the actual classification problem. However, using deep learning models will eliminate this issue by ensuring an end to end training of the model autonomously [13]. Similar approach was put forward by the authors in [14] to target drone detection using DSP along with two Machine Learning algorithms, the Plot Image Learning (PIL) and the K-Nearest Neighbor (KNN). Although the detection ability proved its effectiveness, the

978-1-5386-7747-6/19/$31.00 ©2019 IEEE

overall accuracy of KNN algorithm reported was remarkably low due to the limitation imposed by the design of the solution proposed. Another drawback comes from the fact that PIL requires a large amount of pre-stored images and a consistently varying background noise to avoid biases, which makes deploying it in real environment challenging. Therefore, in this paper, we avoid these restrictions by proposing a method of detecting drones in an environment with a variety of background noises using powerful deep learning techniques in order to support real-time applications.

Deep learning algorithms have proven their effectiveness when being utilized for audio applications, a well-known example of such application is speech recognition [15][16]. As far as we are aware, only one source [17] was found that targets the drone detection problem using drone noise by deep learning techniques, more specifically; the Gaussian Mixture Models (GMM), the Recurrent Neural Network (RNN) and the Convolutional Neural Network (CNN). In their work, the authors have designed and tested the models and concluded that RNN is the best in F1 score among the other two algorithms (Refer to Section III-A for details). Another interesting research conducted by the authors in [18] was the non-verbal audio identification. In their research, they studied, implemented and tested the application of using deep learning techniques for bird species identification. Their study have shown that using deep learning techniques to identify bird species based on their acoustic signatures would yield promising results. Being inspired by their work, in this paper, we aim to design a drone detection and identification solution using deep learning techniques. Namely, the Convolutional Neural Network (CNN), the Recurrent Neural Network (RNN) and the Convolutional Recurrent Neural Network (CRNN). As of the time of writing this paper, no solution was found in the literature that targets both detecting and identifying drones based on their the acoustic fingerprints.

The main contributions of this paper are threefold:

- We investigate and evaluate the effectiveness of the deep learning algorithms in drone detection and identification based on specific evaluation metrics such as accuracy, F1 score, precision and recall.
- We compare the results found to those reported in the literature to provide a better perspective of using deep learning models in drone detection.
- We provide an open-source drone audio dataset to be utilized by the research community to fulfill the shortage of drone training dataset for deep learning models.

The rest of the paper is organized as follows, we describe the dataset, neural networks used and the methodology and setup in Section II. Section III presents the experimental results of the detection and identification methods. Finally, Section IV closes with the conclusion and future work.

II. Proposed Framework

A. Dataset

To satisfy the needs of our proposed method, large amount of drone audio data were required. However, due to various reasons such as privacy, there were no public drone audio data available for this application in literature. Therefore, we have created our dataset by acquiring more than 1300 audio clips of drone sounds and uploaded them to the public which can found in [19]. Moreover, to mimic real life scenarios, we have used the publicly available noise datasets [20] [21] to artificially augment the drone audio clips with noise. The main purpose of the artificial augmentation is to measure the feasibility of the system in a noisy environment. In addition to training the learning algorithm on the augmented sound clips, we have dedicated a portion of the dataset to include pure noise, silence and drone audio clips in order to ensure that the system will be able to detect and identify the drone's sound from similar noises in an environment.

1) Data Acquisition: To acquire the drone sounds, we have recorded the audio clips of the sound generated by the drone's propellers while flying and hovering in a quite indoor environment, this will enable us to publish the dataset publicly in order to be utilized by the research community while ensuring that no privacy or security regulations are being breached. We, also, acquired a balanced number of audio clips per label with equivalent time interval. The importance of this step is to ensure that the audio clips will be equivalently random when fed to the algorithm to avoid any biases. For the purpose of this application, we have recorded a total sound clip of 11 minutes and 6 seconds per drone.

We have decided to use the microphone embedded within a smart-phone for recording the audio clips. Specifically, we have selected the Iphone's integrated microphone for this purpose. This device enabled us to collect audio recordings that are formatted in a CD quality with a sample rate of 44.1KHz, maximum audio bit-rate is 64Kbps with a mono-channel. The audio file is then stored in MPEG-4 audio format (m4a).

2) Data Preprocessing: Preparing the audio files for deep neural networks is a multistage process. The first step is formatting the output audio file produced from the microphone's recording and the background noise clips in order to suit the needs of the application. Therefore, we have reformatted the audio files by converting audio file type, sampling rate, bitrate and the channel as described in Table. I below.

TABLE I
Audio Specs

Sampling rate	Bitrate	Channel Type	Audio format
16KHz	16Kbps	Mono-channel	wav

The second step was to split the formatted audio files into multiple smaller segments by specifying the time intervals at which the audio clip will be segmented, this will enable the deep learning algorithm to learn the features more precisely in comparison to feeding the entire recording at once. Another goal for the segmentation is to optimize the training of the model for real-time deployment in which the time required for the detection and identification is critical. To investigate whether the size of the audio segment would have an effect on the overall performance, we have experimented with multiple segment sizes such as one seconds, two seconds and five second segment. Based on our heuristic observations, we deduced that the one second segmentation outperformed the other time intervals.

While this method may incur loss of some features from the original audio clips, as of the time of writing this paper, the only available off-the-shelf methods to train machine learning or deep learning algorithm on audio input is by converting the audio clips into spectrograms. Various features are then extracted from the generated spectrograms by the algorithm to train the deep learning models. To illustrate the outcome of this process, Fig.1 represents a one second example of a drone's audio. Whereas Fig.2 represents an audio clip of a random noise such as a person typing.

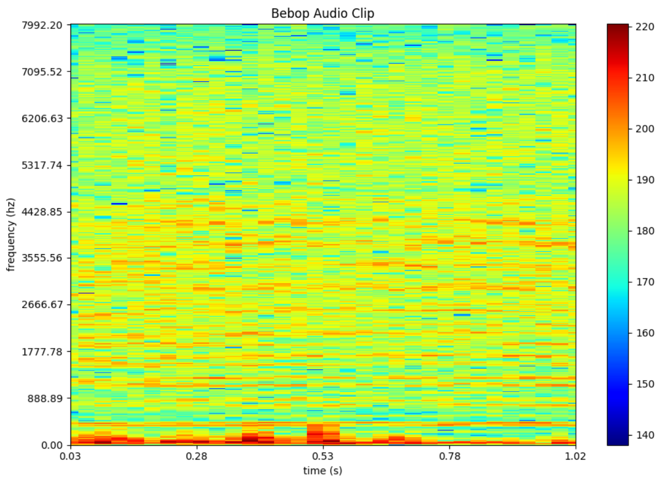

Fig. 1. Example of drone noise in spectrogram representation

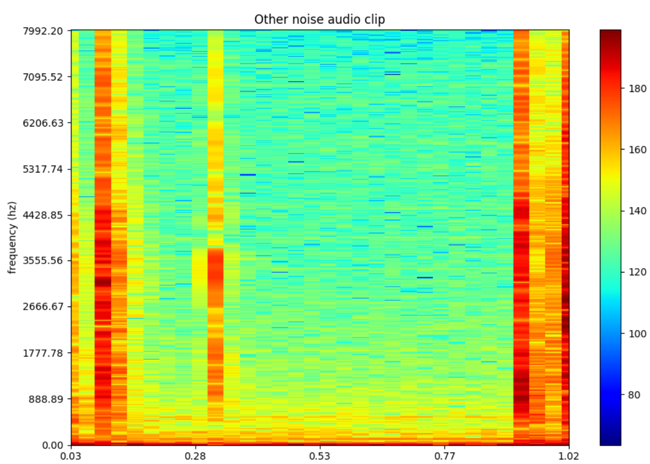

Fig. 2. Example of other noise in spectrogram representation

3) Data Augmentation: Since the application of drone detection and identification could be deployed in areas with a variety of background noises, we have approached the problem by introducing a method of augmentation, in which a real-life background noise is overlapped with the drone audio without any modification on the actual audio features, such as the amplitude or the frequency of the audio clip. Particularly, we have used the background noise from the publicly available dataset[20] [21]. However, it was rather important to reformat the audio clips acquired from these datasets as discussed in Section II-A2 to ensure the consistency of the audio files. Using this mechanism enabled us to mimic real-life scenarios.

4) Data Labeling: We have collected our drone data for two commercially available drones which are the **Bebop** and the **Mambo** drones manufactured by Parrot. Therefore, we labeled our data [19] for the identification problem simply as unknown for other noises in an environment, **Bebop** which represents the first drone and **Mambo** which represents the second drone. The distribution of audio clips acquired per label is represented in Table II.

TABLE II
DATA PER LABEL

Type of Drone	**Records**		
	Original	***Augmented***	***Total***
Bebop	331	335	666
Mambo	331	335	666

Similarly, for the detection section of this system, we have combined the data collected for both Bebop and Mambo drones as one entity and labeled them as **drone** and any other audio clip was labeled as **not a drone**.

B. Neural networks

We have used the open-source code available in [22] for the implementation of each algorithm: CNN, RNN and CRNN, however, we have modified the code slightly to incorporate the termination condition (refer to Section III) in order to suit the needs of this research. The code from [22] was an enhanced version of TensorFlow's simple audio recognition example which is provided as an open-source code [15].

C. Methodology and Setup

We have divided our experiments into two categories, the first is a binary classification experiment in which we target the drone detection solution. We have defined this experiment to handle two use-cases, which are either a drone was detected or no drone in an area. The second category is the multi-class classification experiment which represents the drone identification solution given that we have three distinct labels (Refer to Section II-A4) to identify drones based on their type.

The details of the environment setup at which we deployed the algorithms, trained the models and carried out the experiments are shown in Table III.

TABLE III
ENVIRONMENT SETUP DETAILS

Operating system	Ubuntu 18.04-Linux
Language	Python 2.7
CPU	Intel(R) Xeon(R) x86_64 CPU E5-2695 v4 @ 2.10GHz
Number of CPU	36
Framework and APIs	Google TensorFlow API version 1.11.0

The three main objectives we are attempting to achieve are, firstly, we are interested in investigating the performance of the binary classification aspect on the problem and comparing the results with the literature. Secondly, observing and evaluating the outcome of the multi-class identification classification aspect of the problem and attempting to show the best algorithm out of the three algorithms implemented using different evaluation metrics. Finally, we aim to answer the question of which algorithm has the lowest training and evaluation time.

In order to fulfill the three objectives above, we have chosen to evaluate and compare the different algorithms based on their accuracy, F1 score, precision and recall metrics. Additionally, we have also considered the computational time (CPU time) required to train and test the model as an attribute in evaluating the performance of the models.

We have tried several combinations for the ratio of training to testing datasets and we have deduced from this experiment that the variance of this ratio had minor effect on the overall performance of the model. Therefore, given that the difference is negligible, we opted to use the combination of 70:30 as described in Table IV.

TABLE IV
DATA DISTRIBUTION

Dataset type	Training	Validation	Testing
Percentage	70%	15%	15%

We have also defined the distribution of the labeled data for the binary classification as well as the multiclass identification problem and the learning rate as presented in Table V.

TABLE V
DETAILS ON DATA DISTRIBUTION

Criteria	**Parameter**
Unknown audio files	50%
Learning rate	0.01
Binary Classification Problem	
Drone audio files	50%
Multiclass Classification Problem	
Drone 1 - Bebop	25%
Drone 2 - Mambo	25%

III. EXPERIMENTAL RESULTS

In order to ensure that every algorithm is performing at its optimum, we have carefully chosen the steps below to define the termination condition of the training phase:

1) Executing the algorithm with a very large number of training steps.
2) At an interval of 100 steps, the trained model is tested on the validation-set and the accuracy is calculated and recorded.
3) We compare the new accuracy of the validation-set with the best accuracy achieved so far.
4) If the accuracy did not improve over three successive validation tests, we test the trained model on the testing-set and report the observed results.

Given that the training, validation and testing datasets were shuffled randomly at the start of every execution of learning, we repeated each experiment ten times. Hence, the values discussed in Section III-A and III-B represents the average results of the ten runs.

A. Drone detection: Binary classification results

In the first experiment, we have examined the effectiveness of our proposed system in detecting drones using their acoustic signatures. We have calculated the evaluation metrics which are the accuracy, F1 score, precision and recall for the three different models in addition to the corresponding standard deviation values for the 10 runs as illustrated in Table VI below.

TABLE VI
DETECTION RESULTS

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

It can be deduced from Table VI that CNN have outperformed RNN with a relative improvement of 21.38% in accuracy, 20.32% in F1 score, 20.32% in precision and 27.59% recall. However, the average overall training time required for CNN to yield such precise results was much higher in comparison to RNN. Whereas, it was observed that RNN had the lowest training time and overall performance among the three model. This indicates that there might be a linear correlation between the training time and the performance of the model. Additionally, the performance of CRNN in all evaluation criteria was better than RNN. It is important to take into consideration that the nature of RNN algorithm is best suited for sequential data. Therefore, the shortness in the length of the audio clips might have contributed to the degradation of its performance. Even though, CRNN did not perform better than CNN, the difference between the performance of both models was negligible, in which CNN have shown an improvement of 1.66% in accuracy, 1.98% in F1 score, 1.21% in precision and 1.98% in recall, yet, CRNN was noticeably faster than CNN by 49.07%. This is an interesting finding because it can guide practitioners to consider the model with a lower training time without sacrificing the performance of the model.

The results found in detecting drones using our approach contrasts with the results found in the literature by the authors in [17]. The authors of [17] noted that RNN have achieved the best performance in comparison to CNN. Whereas our results do not support their observation. In fact, we have deduced from our experimental results that CNN have outperformed RNN remarkably. There are a number of factors which might have contributed to the difference of the outcomes between the two approaches such as tuning the algorithm parameters by the authors on the testing-set directly rather than using a validation-set to serve this purpose. Moreover, the discrepancies in our findings can be attributed to the difference of the models' architecture and design parameters such as the number of the convolutional layers used in the CNN algorithms in both applications. Due to the lack of availability of their training and testing datasets, we were not able to perform a direct comparison between the results of both approaches.

Although the results yielded from our proposed experiment does not align with those found by the authors [17], it can nevertheless be concluded that both approaches agreed on the great effectiveness of using deep learning in drone detection using acoustic features.

B. Drone identification: Multiclass classification results

The main goal of our second experiment was to examine the effectiveness of the deep learning methods in identifying drones using their acoustic signatures. We have used the same evaluation metrics mention in Section III-A to investigate the performance of the three models in the multiclass problem, however, the final results were calculated by taking the macro-average over all the classes in the experiment. The overall results of the evaluation metrics are presented in Table VII.

TABLE VII
IDENTIFICATION RESULTS

Evaluation Metric	*RNN*	*CNN*	*CRNN*
CPU-Time (s)	389.02±73.18	807.10±278.09	605.67±252.83
Accuracy (%)	57.16±11.33	92.94±11.89	92.22±1.03
Precision (%)	59.64±13.56	92.75±1.26	92.54±0.95
Recall (%)	57.16±11.27	92.63±1.32	92.23±1.03
F1-score (%)	55.62±13.53	92.63±1.32	92.25±1.01

Results that emerge from this experiment have shown that the results of both CNN and CRNN are outstanding with accuracy, precision, recall and F1 score all above 90%. Moreover, we have observed that CNN have outstandingly outperformed RNN by an improvement of 35.78% in accuracy, 37.01% in F1 score, 33.11% in precision and 35.48% in recall. However, although RNN have shown the worst performance, it was faster than CNN by 51.80% and than CRNN by 35.77% which clearly demonstrates the hypothesis regarding the relation between the computation time and the results of the evaluation metrics mentioned in Section III-A. In addition, it can be observed from the standard deviation values in Table VII that RNN was the fastest to converge regardless of the difficulty of the dataset. Furthermore, it is suspected that the weak performance of RNN algorithm was due to the nature of algorithm since it is mainly based on time-dependent trend which is not the case in this experiment as the audio clips used are of a short length.

Moving on to the comparison between the performance of CNN and CRNN, we have observed that CNN have also performed better than CRNN by 0.72% in accuracy, 0.39% in F1 score, 0.21% in precision and 0.40% in recall. Although CNN have shown some improvement in the performance, one can deduce from the standard deviation values reported in Table VII that the performance of CRNN is more robust in comparison to the other algorithms regardless of the data fed into the algorithm. Moreover, CRNN was significantly faster in execution time than CNN by 24.96%. This finding, as illustrated in Fig. 3, provides a conclusive support for the results found in Section III-A, since in both detection and identification aspects of the problem, it had been observed that practitioners can still utilize a model with significantly fast computational time without jeopardizing the overall precision of the model.

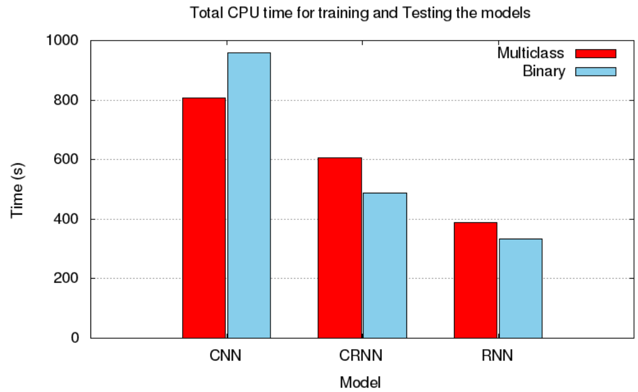

Fig. 3. CPU time Results

In addition to the results presented in Sections III-A and III-B, we have observed that the system was able to identify different drones and other noises while maintaining the precision in the evaluation matrices per label. Table VIII summarizes the average performance of the 10 runs for each label in terms of F1 score for the CNN model. The results presented below suggests that the proposed method has the ability to adjust its identification feature to accommodate more labels based on its application without sacrificing or degrading the performance per label.

TABLE VIII
F1 SCORES PER LABEL

Label	Unknown	Bebop	Mambo
F1 Score	92.607%	93.83%	91.456%

IV. CONCLUSION AND FUTURE WORK

In this paper, we introduced drone detection and identification methods using different deep learning models such as CNN, RNN and CRNN. A key insight into our solution was that the proposed method was able to not only detect the drone presences but also to identify the type of drone. In addition, this approach has shown that the results of the empirical validation of drone detection have supported, in terms of the effectiveness of using deep learning techniques, those previously proposed in the literature. Moreover, from various experiments carried out throughout this paper, we were able to draw a number of conclusions, firstly, that CNN have outperformed RNN and CRNN in terms of accuracy, F1 score, CPU training and testing time, precision and recall in both detecting and identifying of drones. Secondly, since the performance difference between CNN and CRNN were negligible, it can be deduced that CRNN would be a more feasible solution for practitioners as it is remarkably faster than CNN and it had achieved the most stable outcome out of the three algorithms. Finally, this paper has provided the research community with an open-source dataset with various drone audio clips for further analysis.

In our future work will investigate the effect of different parameters on drone detection and identification such as the distance of detecting and identifying drones in an area and combining multiple audio clips from different drones to mimic a swarm of drones attack on physical infrastructure.

Acknowledgment

This publication was supported by Qatar University Internal Grant No. QUCP-CENG-2018/2019-1. The findings achieved herein are solely the responsibility of the authors.

References

[1] J. P. Daniels, *Venezuela's nicolas maduro survives apparent assassination attempt*, Aug. 2018. [Online]. Available: https://www.theguardian.com/world/2018/aug/04/nicolas-maduros-speech-cut-short-while-soldiers-scatter.

[2] *Well-organised gang used drones to deliver drugs to inmates, court told*. [Online]. Available: http://www.itv.com/news/2018-08-30/well-organised-gang-used-drones-to-deliver-drugs-to-inmates-court-told/.

[3] M. Chen, *San carlos woman says drone hovered near bedroom - and wouldn't go away*, 2018. [Online]. Available: https://www.10news.com/news/san-carlos-woman-says-drone-hovered-near-bedroom-wouldnt-go-away.

[4] B. Steffen, *Drone spotted hovering outside bedroom window*, 2018. [Online]. Available: https://www.10news.com/news/drone-spotted-hovering-outside-bedroom-window.

[5] *Drone flies over macron's holiday home in wake of maduro 'attack'*, 2018. [Online]. Available: https://www.thelocal.fr/20180807/drone-flies-over-macrons-holiday-home-in-wake-of-maduro-attack.

[6] Kwqc, *Drone strikes iowa toddler on playground*, Jul. 2018. [Online]. Available: https://wnyt.com/news/drone-strikes-toddler-on-playground-iowa/5010493/.

[7] P. Nguyen, M. Ravindranatha, A. Nguyen, R. Han, and T. Vu, "Investigating cost-effective rf-based detection of drones," in *Proceedings of the 2Nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, ser. DroNet '16, Singapore, Singapore: ACM, 2016, pp. 17–22, ISBN: 978-1-4503-4405-0. DOI: 10.1145/2935620.2935632. [Online]. Available: http://doi.acm.org/10.1145/2935620.2935632.

[8] P. Nguyen, H. Truong, M. Ravindranathan, A. Nguyen, R. Han, and T. Vu, "Matthan: Drone presence detection by identifying physical signatures in the drone's rf communication," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, ACM, 2017, pp. 211–224.

[9] B. Knoedler, R. Zemmari, and W. Koch, "On the detection of small uav using a gsm passive coherent location system," in *Radar Symposium (IRS), 2016 17th International*, IEEE, 2016, pp. 1–4.

[10] Y. Liu, X. Wan, H. Tang, J. Yi, Y. Cheng, and X. Zhang, "Digital television based passive bistatic radar system for drone detection," in *Radar Conference (RadarConf), 2017 IEEE*, IEEE, 2017, pp. 1493–1497.

[11] J. Mezei, V. Fiaska, and A. Molnar, "Drone sound detection," in *2015 16th IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, Nov. 2015, pp. 333–338. DOI: 10.1109/CINTI.2015.7382945.

[12] A. Bernardini, F. Mangiatordi, E. Pallotti, and L. Capodiferro, "Drone detection by acoustic signature identification," *Electronic Imaging*, vol. 2017, no. 10, pp. 60–64, 2017, ISSN: 2470-1173. DOI: doi:10.2352/ISSN.2470-1173.2017.10.IMAWM-168. [Online]. Available: https://www.ingentaconnect.com/content/ist/ei/2017/00002017/00000010/art00009.

[13] N. Takahashi, M. Gygli, and L. V. Gool, "Aenet: Learning deep audio features for video analysis," *CoRR*, vol. abs/1701.00599, 2017. arXiv: 1701.00599. [Online]. Available: http://arxiv.org/abs/1701.00599.

[14] J. Kim, C. Park, J. Ahn, Y. Ko, J. Park, and J. C. Gallagher, "Real-time uav sound detection and analysis system," in *2017 IEEE Sensors Applications Symposium (SAS)*, Mar. 2017, pp. 1–5. DOI: 10.1109/SAS.2017.7894058.

[15] 2018. [Online]. Available: https://www.tensorflow.org/tutorials/sequences/audio_recognition.

[16] X. L. xjli and Z. Z. zixuan, "Speech command recognition with convolutional neural network," 2017.

[17] S. Jeon, J. Shin, Y. Lee, W. Kim, Y. Kwon, and H. Yang, "Empirical study of drone sound detection in real-life environment with deep neural networks," *CoRR*, vol. abs/1701.05779, 2017. arXiv: 1701.05779. [Online]. Available: http://arxiv.org/abs/1701.05779.

[18] E. Sprengel, M. Jaggi, Y. Kilcher, and T. Hofmann, "Audio based bird species identification using deep learning techniques," in *CLEF*, 2016.

[19] S. Al-Emadi, *Saraalemadi/droneaudiodataset*, 2018. [Online]. Available: https://github.com/saraalemadi/DroneAudioDataset.

[20] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*, Brisbane, Australia: ACM Press, Oct. 13, 2015, pp. 1015–1018, ISBN: 978-1-4503-3459-4. DOI: 10.1145/2733373.2806390. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2733373.2806390.

[21] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *CoRR*, vol. abs/1804.03209, 2018. arXiv: 1804.03209. [Online]. Available: http://arxiv.org/abs/1804.03209.

[22] Y. Zhang, N. Suda, L. Lai, and V. Chandra, "Hello edge: Keyword spotting on microcontrollers," *CoRR*, vol. abs/1711.07128, 2017. arXiv: 1711.07128. [Online]. Available: http://arxiv.org/abs/1711.07128.