(POSTER) Drone Detection and Localization using Low-Cost Microphone Arrays and Convolutional Neural Networks

Peteris Racinskis, Janis Arents, Modris Greitans
Robotics and Machine Perception Laboratory
Institute of Electronics and Computer Science, Latvia
peteris.racinskis@edi.lv, janis.arents@edi.lv, modris_greitans@edi.lv

***Abstract*—This paper examines the possibility of using low-cost commercial off-the-shelf audio recording equipment in combination with machine learning techniques to discover the presence of hostile UAVs. A convolutional neural network (CNN) was trained to detect and localize two types of quadrotor drones using ground truth position data collected with motion capture equipment. System performance was evaluated on pre-recorded validation data sets and in real-time operation. In both cases, drones can be successfully detected and localized within the constrained working volumes studied, achieving angular accuracies in the 8-13° range. However, further work remains to be done before system feasibility in outdoor conditions can be established.**

***Index Terms*—CNN, UAV, drone, motion capture, microphone array**

I. Introduction

Unmanned Aerial Vehicles, although seeing combat use as far back as the Second World War [1], have become ubiquitous in the preceding decade, largely thanks to advances in battery and computer technology — enabling the production of low-cost, consumer-grade aircraft controllable from great distances and streaming a high-definition real-time video feed to the operator. Unsurprisingly, military applications of this equipment have been quickly realized as evidenced by the use of commercial off-the-shelf (COTS) quad-rotor drones as artillery spotters, vehicles for explosive payload delivery, and ”suicide” guided munitions [2].

Due to their small size and polymer construction, drones may prove difficult to detect for many types of radar [3]. However, their use of propellers or rotors for propulsion typically makes them noisy and therefore opens up the possibility of acoustic detection and localization. A considerable amount of prior work has been done in aircraft azimuth, position, and velocity estimation using analytic signal processing techniques, hand-crafted filter features, and classical machine-learning algorithms — divided into so-called narrow-band methods [4] which rely on spectral features such as Doppler shift (more useful in velocity estimation), and broadband ones [5], [6] primarily reliant upon signal propagation delay.

Additionally, more recent work has also been done using deep learning to reduce the problem of drone detection to what amounts to a classification task [7]–[9]. Attempts have also been made to use multiple microphone nodes arrayed over a larger area in order to localize drones and estimate their trajectories [10].

One must also note the existence of a variety of proprietary commercial solutions. Some offer detection [11], [12] with a single microphone, whereas others perform localization [13] with multiple microphone arrays — each estimating the target azimuth and their results being combined to produce position estimates. While prior work has been done with drone localization utilizing motion capture as the positional label source [14], this was done using a much larger number of microphones and support vector machines as regression model templates.

In this paper, we establish that an approach utilizing purely learned features is sufficient not only for detection but also localization of drones. Furthermore, we achieve this with few, inexpensive, and readily available sensors, without custom signal processing hardware. Finally, we show that all of this can be done with a neural network simple enough to be trained on a personal computer, without requiring extensive compute resources or large pre-trained models. Our proposed approach, therefore, involves the development of three fundamental subsystems:

- a microphone array — the physical sensor set-up and software interface for extracting data;
- a source of ground-truth position data — some means to generate training data labels;
- a machine learning pipeline — the model architecture, training approach and evaluation metrics.

II. System Overview

We propose using a compact and quickly collected relative position-audio sample pair data set to directly train a parametric model with regression heads for direction and magnitude, as well as a classification head for detection. Inspired by prior work on deep learning methods for sound source localization [15], [16], a convolutional neural network (albeit without residual connections and utilizing only spectral input features) was selected to serve as the regression model. A 2d-to-1d downsampling approach is used on time-frequency inputs to exploit both broadband and narrowband features in the data. An indoor motion-capture system generates the positional data for rapid prototyping and development purposes. The design

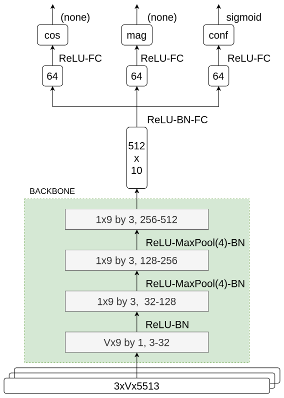

Fig. 1. CNN architecture given a quarter-second sample window at 44.1kHz sample rate and spectrogram view width of *V*.

of the microphone array was deliberately kept simple — with 3 microphones, each within a 3D-printed directional housing and connected to the host device performing inference through a USB sound card, arrayed in a circular planar arrangement.

A. Model Architecture

The model architecture is deliberately minimalist. Its signature can be formally expressed as

$$f_\theta\left(\mathbf{X}_1, ..., \mathbf{X}_V\right) = \left(\hat{\mathbf{r}}, \|\mathbf{r}\|, c\right) \tag{1}$$

$$\mathbf{X} = DFT\left(x(t_1), ..., x(t_N)\right) \tag{2}$$

where $\mathbf{r}$ is the true position vector with respect to the center of the microphone array, hence $\hat{\mathbf{r}}$ and $\|\mathbf{r}\|$ are its direction and magnitude components respectively; c is a binary classifier output. The V input vectors $\left(\mathbf{X}_1, ..., \mathbf{X}_V\right)$ (”slices”), constitute a real-valued spectrogram. V itself is referred to as spectrogram view width or simply view width.

Each slice is computed as the Discrete Fourier Transform (DFT) of N amplitude observations $\left(x(t_1), ..., x(t_N)\right)$ taken at a sampling frequency f_{sample} over a sampling period T. All inputs have real values, so only positive frequency values $X_{i \leq \frac{N}{2}}$ of the transform need to be considered. The V slices that together form the input of the model are themselves collected at a frequency of $f_{slice} = 180\text{Hz}$, which is not related to the audio sampling frequency $f_{sample} = 44.1\text{kHz}$.

For inference, the absolute value of the transform is used. These were normalized by $\log_{10}\left(|X_i|^2\right)$ and had the dominant lowest frequency components truncated to improve model performance. Each spectrogram slice $\mathbf{X}$ for each channel is furthermore individually standardized, to eliminate total loudness differences between channels. The model is parametrized by view width V and sample period length T. The latter was fixed at 0.25s for all experiments, resulting in the input data shape illustrated in figure 1 when combined with the constant sampling frequency of 44.1kHz.

The downsampling convolutional part of the network is a single 2-dimensional layer followed by a sequence of 1-dimensional ones. All convolutional layers use a ReLU activation. Prediction heads have a single hidden layer with a ReLU activation. The direction output $\hat{\mathbf{r}}$ and magnitude output $\|\mathbf{r}\|$ have no activation, whereas the classifier value c has a sigmoid nonlinearity applied to it.

The model was implemented using *PyTorch*, and the training loss could be expressed as

$$\begin{aligned}\left(\mathcal{L}_{cos}(\hat{\mathbf{r}}, \hat{\mathbf{r}}_{true}) + \mathcal{L}_{mag}(\|\mathbf{r}\|, \|\mathbf{r}\|_{true})\right) * c_{true} + \\ + \mathcal{L}_{cls}\left(c, c_{true}\right)\end{aligned} \tag{3}$$

with $\mathcal{L}_{cos}, \mathcal{L}_{mag}$ being Huber losses for the direction (cosine) and magnitude outputs respectively, and $\mathcal{L}_{cls}$ being the cross entropy classification loss. Regression losses are masked by the label class c_{true} (0 when no drone is present in the sample)

B. Audio Data Collection

A 3-element planar microphone array was implemented, leaving room for an additional fourth input in case this proved to be necessary for direction finding outside of the horizontal plane. A directional housing was 3d printed for each microphone, attached to a central fixture using variable-length spans of 30mm modular aluminum profile. An audio playback system was added to the data-gathering set-up to allow for the augmentation of training data with various noise types. It was found that including in particular sharply punctuated, wide-band noise sources (e.g., clapping) practically eliminated observed model tracking of noise in real-time inference.

When collecting training or test data, sound recordings are made in chunks and stored alongside position time-series data. The Fourier conversion, stacking, and normalization steps are handled when loading this data into memory. The end use case, however, involves working with real-time audio streams. For this purpose, a queue of sample chunks is maintained, on which a sequence of DFTs is periodically computed to produce spectrograms of the desired width.

C. Relative Position Data Collection

Training labels — ground truth positions of the drone w.r.t. the microphone array — were collected indoors, using *OptiTrack* motion capture equipment and *Motive* software, within a 6 × 6 × 3 work area, as in [17]. Two consumer-grade drones were used — a *Syma X5HW* and *DJI Phantom*. Sound recordings were made in short chunks synchronized with sequences of drone positions at f_{slice}, and linear interpolation was employed to produce an evenly spaced time series. Samples are only appended to the data set at intervals where a sufficient change in the drone's position has been observed to reject any made when the drone is landed or otherwise immobilized.

III. Results

A. Experimental Setup

A training data set consisting of approximately 26 minutes of recordings with each drone was collected. Classifier performance was improved by making sure to structure recordings symmetrically — including a similar amount of data collected under the same ambient conditions (noise type) both with and without the drone. Validation data sets used for debugging and controlling the training process were created in separate recording sessions and stored separately. These are 4×180 samples long for each type of drone, with/without the drone present and with/without additional noise. To aid in evaluating and debugging model performance, visualization tools focusing primarily on a top-down view of the working volume were developed, both for use with a prerecorded data set and running inference in real time.

B. Evaluation Metrics and Quantitative Results

5 metrics were computed to evaluate model performance on validation data sets.

$$\epsilon_\theta = \deg\left(\overline{\arccos\left(\hat{\mathbf{r}} - \hat{\mathbf{r}}_{true}\right)}\right) \tag{4}$$

$$\epsilon_{\|r\|} = \overline{(|\|\mathbf{r}\| - \|\mathbf{r}\|_{true}|)} \tag{5}$$

$$\epsilon_r = \overline{(|\|\mathbf{r}\| \cdot \hat{\mathbf{r}} - \mathbf{r}_{true}|)} \tag{6}$$

$$\epsilon_{\|r\|\%} = \overline{\left(\frac{|\|\mathbf{r}\| - \|\mathbf{r}\|_{true}|}{\|\mathbf{r}\|_{true}}\right)} * 100 \tag{7}$$

$$\epsilon_{r\%} = \overline{\left(\frac{\|\mathbf{r} - \mathbf{r}_{true}\|}{\|\mathbf{r}\|_{true}}\right)} * 100 \tag{8}$$

being mean angular error, absolute magnitude error, absolute positional error, relative magnitude and relative positional error respectively.

Table I shows best-attained in each metric for models trained (and evaluated) on data containing each type of drone as well as both at once, the latter illustrating a clear capability to learn multi-modal distributions. Table II shows the degradation experienced when exposed to a type of drone not seen in the training data. Notably, the *Syma* drone is significantly quieter, leading models trained only on it to underestimate distances when faced with its louder counterpart and vice versa.

IV. Conclusions

Our work successfully demonstrates that a CNN simple enough to be trained on a personal computer, alongside a spartan sound recording setup can be used to successfully detect and localize quadrotor drones with an angular error in the 10-degree range, using approximately 52 minutes of synchronized audio-position data from a motion capture system. However, more work remains to be done to verify if this approach is feasible in more noisy outdoor conditions, and models show weaknesses when exposed to unfamiliar target types.

TABLE I
Highest performance by drone used

Drone type	**Evaluation metric**				
	ϵ_θ	$\epsilon_{\|r\|}$	ϵ_r	$\epsilon_{\|r\|\%}$	$\epsilon_{r\%}$
DJI	8.51°	0.18m	0.43m	8.29%	19.19%
Syma	12.86°	0.22m	0.60m	11.43%	28.17%
Both	10.06°	0.22m	0.49m	10.59%	22.94%

TABLE II
Highest generalization performance

Drone type	**Evaluation metric**			
	ϵ_θ	$\epsilon_{\|r\|\%}$	$\epsilon_{\hat{r}\%}$	cls%
DJI → Syma	25.70°	34.52%	63.07%	99.77%
Syma → DJI	27.24°	30.84%	63.70%	95.49%

Acknowledgment

This research was partially supported by Latvian state research program No. VPP-EM-FOTONIKA-2022/1-0001 ”Viedo materiālu, fotonikas, tehnoloģiju un inženierijas ekosistēma”

References

[1] Blackwelder, Major Donald I. ”The long road to Desert Storm and beyond: the development of precision guided bombs”. Pickle Partners Publishing, 2015.

[2] BBC. ”How are ’kamikaze’ drones being used by Russia and Ukraine?” bbc.com Accessed: 13/01/23 URL: https://www.bbc.com/news/world-62225830

[3] Coluccia, Angelo, Gianluca Parisi, and Alessio Fascista. ”Detection and classification of multirotor drones in radar sensor networks: A review.” Sensors 20, no. 15 (2020): 4172.

[4] K. W. Lo and B. G. Ferguson, “ Tactical unmanned aerial vehicle localization using ground-based acoustic sensors,” in Proceedings of Intelligent Sensors, Sensor Networks and Information Processing Conference, ISSNIP (2004), pp. 475–480.

[5] Kaplan, Lance M., and Qiang Le. ”On exploiting propagation delays for passive target localization using bearings-only measurements.” Journal of the Franklin Institute 342, no. 2 (2005): 193-211.

[6] Kozick, Richard J., and Brian M. Sadler. ”Source localization with distributed sensor arrays and partial spatial coherence.” IEEE Transactions on Signal Processing 52, no. 3 (2004): 601-616.

[7] Casabianca, Pietro, and Yu Zhang. ”Acoustic-based UAV detection using late fusion of deep neural networks.” Drones 5, no. 3 (2021): 54.

[8] Al-Emadi, Sara, Abdulla Al-Ali, and Abdulaziz Al-Ali. ”Audio-based drone detection and identification using deep learning techniques with dataset enhancement through generative adversarial networks.” Sensors 21, no. 15 (2021): 4953.

[9] Fang, Jian, 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, no. 2 (2022): 968-981.

[10] Yang, Bowon, Eric T. Matson, Anthony H. Smith, J. Eric Dietz, and John C. Gallagher. ”UAV detection system with multiple acoustic nodes using machine learning models.” In 2019 Third IEEE International Conference on Robotic Computing (IRC), pp. 493-498. IEEE, 2019.

[11] Prime Consulting & Technologies. ”Acoustic sensors” anti-drone.eu Accessed: 13/01/23 URL: https://anti-drone.eu/products/acoustic-sensors/

[12] Roke. ”Acoustic drone detection” roke.co.uk Accessed: 13/01/23 URL: https://roke.co.uk/innovations/acoustic-drone-detection

[13] Squarehead Technology. ”Defense” sqhead.com Accessed: 13/01/23 URL: https://www.sqhead.com/drone-detection/

[14] Baron, Valentin, Simon Bouley, Matthieu Muschinowski, Jerome Mars, and Barbara Nicolas. ”Acoustic localization and identification of drones with a disturbance source.” In Forum Acusticum 2020, pp. 3149-3154. 2020.

[15] Yalta, Nelson, Kazuhiro Nakadai, and Tetsuya Ogata. ”Sound source localization using deep learning models.” Journal of Robotics and Mechatronics 29, no. 1 (2017): 37-48.

[16] E. L. Ferguson, S. B. Williams and C. T. Jin, ”Sound Source Localization in a Multipath Environment Using Convolutional Neural Networks,” 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 2018, pp. 2386-2390, doi: 10.1109/ICASSP.2018.8462024.

[17] Racinskis, Peteris, Janis Arents, and Modris Greitans. 2022. ”A Motion Capture and Imitation Learning Based Approach to Robot Control” Applied Sciences 12, no. 14: 7186. https://doi.org/10.3390/app12147186