DDL: A DATASET FOR DRONE DETECTION AND LOCALIZATION FROM MULTI-CHANNEL AUDIO AND A DEEP UNCERTAINTY-AWARE FRAMEWORK

Saeid Safavi[†] *Özkan Çaylı*[†] *Mohammad Amin Safavi*[‡] *Ben Cook*[⊕] *Wenwu Wang*[†]

[†] Centre for Vision, Speech and Signal Processing, University of Surrey, United Kingdom
[‡] DeepX Inc., Japan
[⊕] Airspeed Electronics Ltd., United Kingdom

ABSTRACT

Due to the widespread use of drones in an urban environment, drones present an increased risk to the safety of urban life. Reliable detection of drones becomes crucial for countering the hazard introduced by drones. However, drones are difficult to detect because of their size and customization. This paper introduces DDL, a dataset aimed at drone sound detection, classification, and localization via a specially constructed set of microphones. As a baseline, we propose a deep uncertainty-aware framework implementing Conformer for joint drone classification and localization. We employ heteroscedastic loss functions that jointly estimate means and variances for spatial localization to model prediction uncertainty. Experiments on the DDL dataset demonstrate a classification accuracy of 99.9% and a Euclidean distance mean absolute error (MAE) of approximately 16 meters. The uncertainty estimates are well-calibrated, with coverage closely matching the expected confidence intervals (68%, 95%, and 99.7%) as defined by the empirical rule, suggesting DDL as a benchmark dataset for audio-based drone localization.

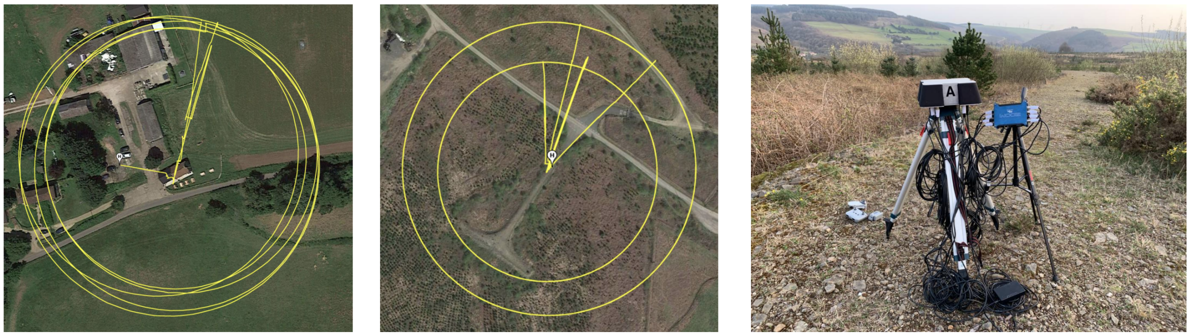
***Index Terms*—** dataset, drone localization, multi-channel audio, uncertainty estimation, Conformer, DOA

1. INTRODUCTION

Drones, also known as unmanned aerial vehicles (UAVs), have gained popularity in recent years due to their ability to fly autonomously. Autonomous human transport machines and delivery systems for postal services are among the applications that have been proposed [1]. In addition, search and rescue scenarios involving individuals in emergency circumstances who must be located swiftly in difficult-to-reach locations represent a potentially substantial field of application. Although video cameras have been used to construct a variety of UAV-embedded solutions to deal with such scenarios [2, 3], several other approaches have been explored. Apart from the useful applications of drones, the growing popularity of commercial hobby drones poses unanticipated hazards to the environment in which we live, such as fear for people or potential damage to critical infrastructure. A typical four-propeller drone is excellent for use in leisure and broadcasting, but it also makes current defensive systems look like antiquated legacy technologies. Some mishaps have previously shown that these drones can easily breach the highest level of security, such as drones flying near Gatwick airport back in 2018, which caused a two-day airport closure [4]. As a result, the ability to detect the presence of a drone is of the utmost importance to avoid any danger.

Most publicly available drone sound datasets contain a small number of recordings and are not specifically designed for drone detection and localization tasks. As a solution, this study introduces a new dataset, dubbed Drone Detection and Localization (DDL) to advance research into audio properties during UAV flights and the development of new drone sound detection and localization methods. Two types/sizes of drones were used: DJI Phantom 4 Pro and DJI Mini 2. To capture drone sounds, a receiver with an 8-channel cube-shaped microphone array was carefully designed. To detect and localize drones from the microphone array recordings, we leverage recent progress in machine learning-based methods, such as those for audio classification, e.g., acoustic scene classification and Direction of Arrival (DOA) estimation in a real-world setting. Early audio classification systems have considered Mel-frequency cepstrum coefficients [5] with Gaussian mixture models [6]. Recent research has focused on using Deep Neural Networks for ambient sound recognition, such as Convolutional Neural Networks (CNNs) [7], Recurrent Neural Networks (RNNs) [8], and Transformer-based architectures [9]. Despite the fact that extensive research has been conducted for audio classification, little has been done for drone sound localization. In drone sound localization, uncertainty modeling is crucial for building reliable systems, as the audio environment is often noisy and unpredictable. Distinguishing between confident and uncertain predictions improves decision-making under ambiguity and enhances system robustness. Conventional methods such as Bayesian neural networks [10] and Monte Carlo dropout [11] model epistemic uncertainty but are computationally expensive. More scalable approaches, such as deep ensembles [12],

979-8-3503-2411-2/25/$31.00 ©2025 IEEE



(a) (b) (c)

Fig. 1: (a) DJI Mini2 Orbits, Guilford, UK, (b) DJI 4 Pro Orbits, Port Talbot, UK, and (c) recording setup

improve robustness but require multiple models. However, heteroscedastic regression [13] predicts both the mean and variance of the results in a single forward pass.

This paper presents a dataset for drone sound classification and localization. Additionally, we propose a framework that jointly performs drone classification and localization using deep learning models while modeling aleatoric uncertainty, which is particularly useful for handling environmental noise in outdoor drone localization scenarios, through a heteroscedastic loss.

2. DATASET DESCRIPTION

This section first reports the hardware used in data collection; secondly, we describe the recording scenes and session design; and thirdly, we explain the dataset creation process and distributions.

2.1. Drones and recording setup

There are two drone types used as sound sources in this dataset, namely DJI Mini 2 and DJI Phantom 4 Pro. These drones were chosen due to their differences in size, auto-pilot capabilities, and ease of access to their rich flight logs. Both drones have similar sensor suites, flight logs, and auto-pilot capabilities, making the location data more homogeneous and easier to use, while their differences in size and motor types increase the diversity of the recorded data. The other useful feature provided by the chosen drones was their autopilot capability. The autopilot feature was used in both drones to automate the recording of flight trajectories. This enables the data collected to have a more uniform distribution and timing compared to a remotely operated drone. The DJI flight logs provide reliable GPS data, velocity vector, and time stamps which will be used to localize and synchronize position and audio of drones with respect to the microphone array, respectively. For the recording setup, a microphone array with eight channels is used. The microphone heads are located at the four corners of the box in pairs. The microphone array is designed to be waterproof and is equipped with windjammers. Each microphone head is connected to a field recorder using XLR cables. For multi-track recording, Zoom F8N is used to capture audio at 96kHz sample rate, resulting in a synced 8-channel .wav output file.

2.2. Data collection and recording scene design

The following elements in the scene were taken into account: 1) different open environments, 2) a diverse set of drone trajectories with respect to (w.r.t) the microphone array. The recording sessions are located in Port Talbot and Guildford, UK. The former is a quiet rural environment with less background noise, and the latter is a suburban environment with more background noise, such as cars passing and residents chatting. There are 12 recording sessions in total, and each session is approximately 10-22 minutes, a variable dictated by drone battery capacity. Two of the recording sessions exclusively record the environment noise for 10 minutes. In each session, the drone starts a sequence of trajectories using the autopilot. The drone first flies to a starting point of an orbital path with a fixed radius and hovers in that position, waiting for confirmation to start the orbital path. The recording process begins as the drone receives confirmation. In addition, to ensure the synchronization of audio and spatial data, an audio clue (digital beep sound) was added to mark the start of the movement. Later, the audio clue and flight log data were combined to synchronize the timestamps of the spatial and audio data. Satellite images of drone trajectories and scene setup are shown in Fig. 1. The orbital paths are two-dimensional circles with the microphone array in the center and a constant altitude of 30 meters above the ground. The velocity profiles of the orbital paths are defined to have a uniform amount of audio samples for each degree. However, this was not guaranteed due to wind blows, autopilot errors, and drone acceleration/deceleration at the beginning/end of the path. For greater distances, samples are taken from a single degree with the drone hovering in a fixed radius since performing orbits was

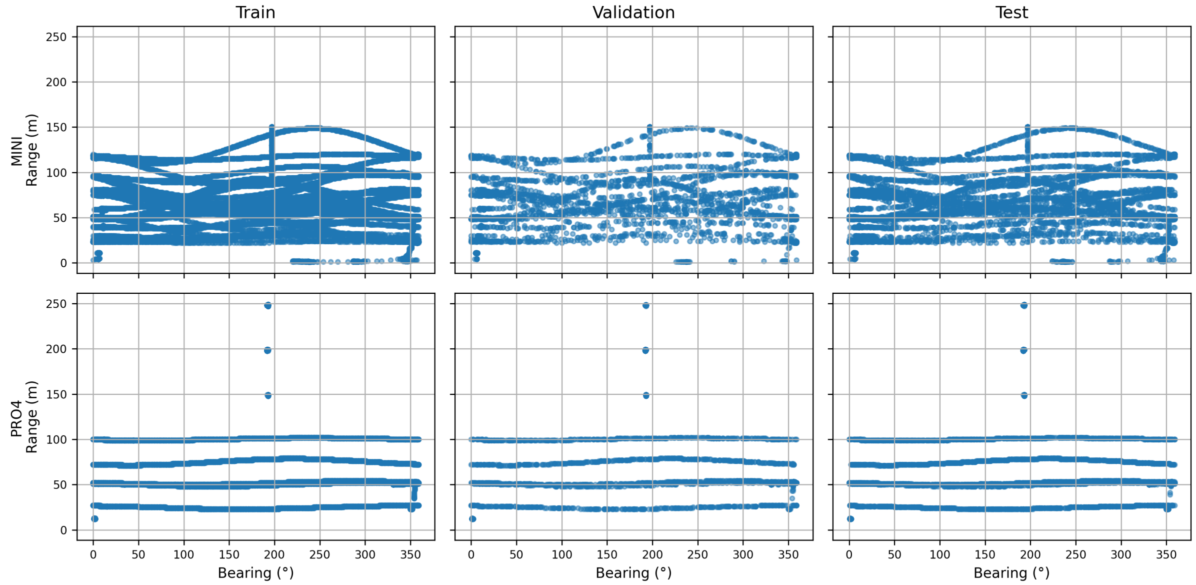


Fig. 2: Bearing vs. Range distribution of samples for each drone type (rows) across Train, Validation, and Test sets (columns).

Table 1: Transposed distribution of drone samples across distance bins for MINI and PRO4 models.

Distance	MINI				PRO4			
	Train	Validation	Test	Overall	Train	Validation	Test	Overall
20m	1060	150	300	1510	840	113	257	1210
40m	2912	424	834	4170	2607	384	750	3741
60m	5967	899	1691	8557	4959	679	1418	7056
85m	5974	820	1695	8489	2514	383	723	3620
110m	4843	678	1371	6892	2650	383	766	3799
175m	3398	480	1011	4889	709	115	189	1013
225m	0	0	0	0	1269	186	354	1809
275m	0	0	0	0	1007	122	274	1403
Sum	24154	3451	6902	34507	16555	2365	4731	23651
Percentage (%)	41.53	5.93	11.87	59.33	28.47	4.07	8.13	40.67

not feasible due to potential collisions. A breakdown of the sample count and distributions for each drone at different distances is presented in the next section.

2.3. Data pre-processing and statistics

The pre-processing consists of the following operations. First, the spatial data of drones w.r.t the microphone array are extracted from the flight logs. Second, audio and spatial time stamps are synchronized, and last, audio recordings are sampled and annotated with the relevant metadata.

In the first step, Haversine and forward azimuth formulas are used to calculate the distance and bearing (angle of arrival) between the two GPS coordinates. The Haversine formula and forward azimuth are given as follows [14]:

$$\begin{aligned} a &= \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1\cos\phi_2\sin^2\left(\frac{\Delta\lambda}{2}\right) \\ c &= 2\cdot\operatorname{atan2}\left(\sqrt{a},\sqrt{1-a}\right) \\ d &= R\cdot c \end{aligned} \tag{1}$$

where ϕ is latitude, λ is longitude, and R is earth's radius (mean radius = 6,371 km).

$$\begin{aligned} \theta = \operatorname{atan2}\Big(&\sin(\Delta\lambda)\cos\phi_2, \\ &\cos\phi_1\sin\phi_2 - \sin\phi_1\cos\phi_2\cos(\Delta\lambda)\Big) \end{aligned} \tag{2}$$

where ϕ_1, λ_1 is the start point, and ϕ_2, λ_2 the end point ($\Delta\lambda$ is the difference in longitude). The other metadata taken from the flight logs is the date, UTC, drone model, drone altitude, and ambient temperature. As for the second step, i.e., time synchronization, the goal is to calibrate the timestamps of the flight log generated by the drone and the timing of the audio recordings. This is an essential step, as it affects the labeling of audio samples. A small delay of one second between drone position data and audio samples can cause a change in data distribution and degrade quality and reliability of the dataset. In order to have an accurate synchronization, multiple parameters are taken into account. Initially, the UTC timestamp of every recording session is checked with the flight log. Then, the audio cues are identified from the recordings, and their timestamp in UTC is extracted. The audio cue UTC values are then compared with the flight log data, and the timestamps are calibrated. It is easy to find the moments at which the drone is hovering and waiting for the signal to start its trajectory in the flight log. The audio and flight logs are trimmed and calibrated using location data and velocity. After the audio and flight logs are calibrated and trimmed, the audio files are stored with a length of 100 milliseconds and annotated with the metadata and spatial labels from the flight logs. It should be noted that the samples do not overlap. Each sample is an 8-channel 100ms-long audio file named in a particular convention to make it easier to parse and be used in data loaders. The dataset and the aforementioned naming convention are available online on Zenodo [1]. The data distribution for an orbital trajectory is shown in Fig. 2, where the bearing represents the DOA. It can be seen that the smaller drone, DJI Mini 2, was not as accurate as DJI Phantom Pro 4 in path tracking, as shown in the scatter plots. The type of drone data recorded, distances from sources, and the number of samples for both drones are depicted in Table 1.

3. DRONE SOUND DETECTION AND DOA ESTIMATION

Our proposed deep uncertainty-aware framework for drone detection and localization consists of three main stages: feature extraction, neural network-based drone classification and localization, and uncertainty modeling. The framework utilizes log-mel spectrogram features derived from multi-channel audio signals captured by an 8-microphone array. Three neural network architectures— Audio Spectrogram Transformer (AST), Conformer, and Convolutional Recurrent Neural Network (CRNN)—were independently trained and evaluated within this framework to assess their effectiveness for drone classification and localization.

The input data consists of multi-channel audio recordings captured using an 8-channel microphone array, sampled at 96 kHz and segmented into 0.1-second audio clips (9600 samples each). From these recordings, log-mel spectrograms with 64 mel frequency bands were extracted for each channel, resulting in feature tensors of shape $8 \times 64 \times T$, where T represents the temporal dimension determined by the audio segmentation and processing parameters. The log-mel spectrogram features effectively capture drone-specific acoustic signatures, enabling accurate classification and localization within the framework. The targets are scaled to enable stable learning: The bearing angle, originally defined between $0°$ and $360°$, is shifted to $[-180°, 180°]$ and represented using its sine and cosine components, both constrained to the range $[-1, 1]$. During prediction, the sine and cosine means are constrained using a $\tanh$ activation. The range, originally measured between 0 and 250 meters, is linearly scaled to the $[0, 1]$ interval. During prediction, the mean of range is constrained using an activation sigmoid. Each model processes the input spectrograms and produces a feature representation that is passed through a shared prediction module. The shared prediction head outputs three types of information: a binary classification logit indicating the type of drone class, the mean and logarithmic variance values for the sine and cosine components of the bearing, and the mean and logarithmic variance for the scaled range.

Although the backbone architectures differ, they are adapted into the framework as follows: for AST, the input spectrogram is patched and projected into an embedding space. A Transformer encoder processes the sequence of patches, and the representation corresponding to the classification ([CLS]) token is used as input to the prediction head. For Conformer, a convolutional frontend first extracts local representations, which are then linearly projected and passed through a stack of Conformer layers to model temporal dependencies. Global average pooling across the temporal dimension generates the feature vector fed into the prediction head. For CRNN, a convolutional encoder extracts features, which are flattened and processed through a bidirectional GRU to capture temporal patterns. The outputs are globally average pooled across time before being passed into the prediction head.

Algorithm 1 Training and Prediction with Conformer

```
Require: Mini-batch $\mathbf{X} \in \mathbb{R}^{B \times 8 \times 64 \times T}$
Ensure: Class logit $z$, sine mean $\mu_{\sin}$, sine log-variance $\log \sigma^2_{\sin}$, cosine mean $\mu_{\cos}$, cosine log-variance $\log \sigma^2_{\cos}$, range mean $\mu_r$, range log-variance $\log \sigma^2_r$
for each mini-batch $\mathbf{X}$ do
    $\mathbf{F} \leftarrow \text{CNN}(\mathbf{X})$
    $\mathbf{S} \leftarrow \text{Project}(\mathbf{F})$
    $\mathbf{H} \leftarrow \text{Conformer}(\mathbf{S})$
    $\mathbf{h} \leftarrow \text{GlobalAvgPool}(\mathbf{H})$
    $\{z, (\mu_{\sin}, \log \sigma^2_{\sin}), (\mu_{\cos}, \log \sigma^2_{\cos}), (\mu_r, \log \sigma^2_r)\} \leftarrow \text{PredictionHead}(\mathbf{h})$
end for
for each target $(y_{\text{class}}, y_{\sin}, y_{\cos}, y_r)$ do
    $\mathcal{L}_{\text{cls}} \leftarrow \text{BCE}(z, y_{\text{class}})$
    $\mathcal{L}_{\sin} \leftarrow 0.5 \times \exp(-\log \sigma^2_{\sin}) \times (y_{\sin} - \mu_{\sin})^2 + 0.5 \times \log \sigma^2_{\sin}$
    $\mathcal{L}_{\cos} \leftarrow 0.5 \times \exp(-\log \sigma^2_{\cos}) \times (y_{\cos} - \mu_{\cos})^2 + 0.5 \times \log \sigma^2_{\cos}$
    $\mathcal{L}_r \leftarrow 0.5 \times \exp(-\log \sigma^2_r) \times (y_r - \mu_r)^2 + 0.5 \times \log \sigma^2_r$
end for
$\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{cls}} + \mathcal{L}_{\sin} + \mathcal{L}_{\cos} + \mathcal{L}_r$
Update model parameters using backpropagation with $\mathcal{L}_{\text{total}}$
```

To capture uncertainty, the framework employs a heteroscedastic modeling approach for both bearing and range estimation. For each target variable (sine of bearing, cosine of bearing, and range), the models predict both a mean and a log-variance value. The loss function used for training combines binary cross-entropy (BCE) for drone classification and negative log-likelihood (NLL) for each regression target,

[1] https://doi.org/10.5281/zenodo.6459182

assuming Gaussian distributions with input-dependent variances. Formally, given a prediction mean μ and log-variance $\log\sigma^2$ for a target y, the heteroscedastic regression loss is defined as:

$$\mathcal{L}_{\text{het}} = 0.5 \times \exp(-\log\sigma^2) \times (y-\mu)^2 + 0.5 \times \log\sigma^2 \quad (3)$$

This allows the model to express higher uncertainty for ambiguous or noisy inputs. The total loss used for training is the sum of the classification loss and the heteroscedastic regression losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\sin} + \mathcal{L}_{\cos} + \mathcal{L}_r \quad (4)$$

where $\mathcal{L}_{\text{cls}}$ is the binary cross-entropy loss for classification, and $\mathcal{L}_{\sin}$, $\mathcal{L}_{\cos}$, and $\mathcal{L}_r$ are the heteroscedastic losses for sine, cosine, and range predictions, respectively.

All neural network models within the framework were trained using an NVIDIA A100 GPU. Training involved optimizing a combined loss function consisting of BCE and heteroscedastic regression losses, as outlined in Algorithm 1. Optimization was performed using the Adam optimizer, with a learning rate of 1×10^{-4}, and a batch size of 32.

4. EXPERIMENTS AND RESULTS

Table 2: Prediction and uncertainty performance on the test set.

Prediction Metrics	**AST**	**Conformer**	**CRNN**
# of Parameters	33,285,127	2,575,687	7,915,911
Accuracy (%)	97.59	99.93	99.94
Precision (%)	97.51	99.87	99.87
Recall (%)	96.56	99.96	99.98
F1 Score (%)	97.03	99.92	99.93
Range MAE (m)	16.33	5.20	4.77
Bearing MAE (°)	31.89	11.44	11.78
Euclidean MAE (m)	43.43	16.05	15.76
Uncertainty Metrics			
Coverage@1σ (%)	72.97	56.20	50.52
Coverage@2σ (%)	95.42	85.47	79.55
Coverage@3σ (%)	99.05	97.12	90.90
Within 1σ (%)	72.97	56.20	50.52
1σ–2σ (%)	22.45	29.28	29.03
2σ–3σ (%)	3.63	11.64	11.35
Outside 3σ (%)	0.95	2.88	9.10
Mean Harmonic Var (Angle)	0.21726	0.02981	0.01142
Mean Harmonic Var (Range)	0.32578	0.04466	0.01712

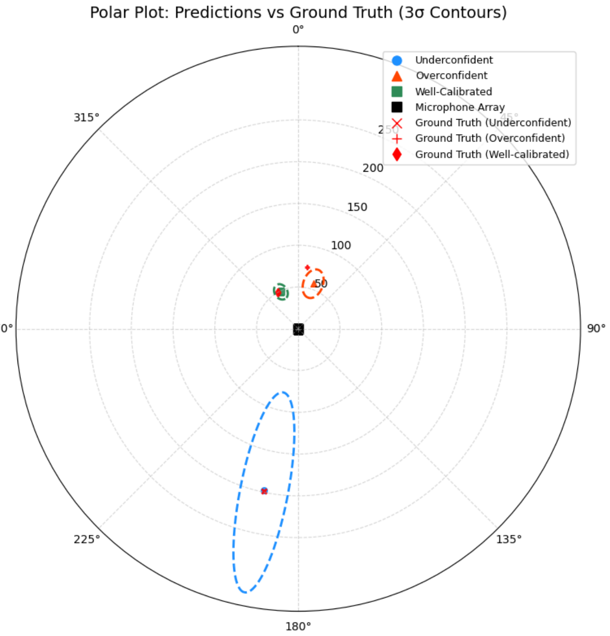


Fig. 3: Polar plot showing three example predictions with 3σ uncertainty contours: well-calibrated (green), overconfident (orange), and underconfident (blue). Ground truth locations are marked with red markers.

In this section, we present the experimental results obtained from independently training and evaluating three neural network architectures AST, Conformer, and CRNN within the proposed framework. Evaluation was performed on a test set comprising 11,632 real-world multi-channel drone audio recordings. Model performance was assessed in terms of classification accuracy, precision, recall, F1-score, localization error (range, bearing, and Euclidean MAE), and uncertainty estimation metrics.

Table 2 summarizes the results for model complexity, prediction performance, and uncertainty evaluation. In terms of classification and localization performance, CRNN achieved the highest overall classification accuracy (99.94%) and the lowest range and Euclidean localization errors (4.77 m and 15.76 m), respectively. Conformer followed closely with no significant difference compared to CRNN in localization errors. In contrast, AST demonstrated substantially higher localization errors, with a Euclidean MAE of 43.43 m, indicating limited suitability for precise localization tasks. For uncertainty estimation, AST achieved coverage closest to the theoretical 68-95-99.7% rule expected under a Gaussian assumption, outperforming the other architectures in terms of coverage consistency. Here, coverage refers to the proportion of test cases where the ground truth position falls within the predicted confidence intervals derived from the mean and standard deviation outputs of the model. Conformer ranked second in uncertainty coverage, while CRNN exhibited poorer calibration, with 9.10% of the predictions falling outside the 3σ confidence interval. However, despite the strong coverage of AST, its predicted uncertainty variances were significantly higher than those of Conformer and CRNN, reflecting a broader spread and lower sharpness of the predictive distributions. This high variance, combined with its poorer localization performance, limits the practical reliability of AST in this framework. Conformer provided the best balance across all aspects: achieving high classification and localization performance, reliable uncertainty calibration, and the smallest model size with approximately 2.6 million parameters, compared to 7.9 million for CRNN and 33 million for AST. This highlights Conformer-based architecture as the most suitable choice among the three for drone classification and localization with the proposed framework. The uncertainty estimations in Figure 3 show how the predicted confidence regions relate to the true error. The dashed contours represent 3σ confidence regions. The well-calibrated prediction shows uncertainty that aligns with the actual difference between the predicted and ground-truth positions, whereas the underconfident and overconfident predictions reflect poor calibration by overestimating or underestimating the uncertainty of the model. To the best of our knowledge, this is the first work to incorporate and visualize localization uncertainty in the context of drone localization on an acoustic dataset, marking a novel contribution to the field.

5. CONCLUSION

We have introduced a novel audio dataset for drone detection and localization. The dataset comprises real drone audio recordings captured with an 8-channel microphone array and supports both classification and localization tasks. In addition to the dataset, we propose a unified deep learning framework that jointly performs classification and localization while modeling input-dependent aleatoric uncertainty through a heteroscedastic loss. Three baseline architectures—AST, Conformer, and CRNN—are implemented and evaluated within this framework. Experimental results demonstrate that the proposed framework achieves accurate classification and localization performance, with the Conformer backbone performing the best, suggesting the effectiveness of the dataset as a benchmark.

6. REFERENCES

[1] R. D'Andrea, "Guest editorial can drones deliver?," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 3, pp. 647–648, 2014.

[2] L. Lopez-Fuentes, J. van de Weijer, M. González-Hidalgo, H. Skinnemoen, and A. Bagdanov, "Review on computer vision techniques in emergency situations," *Multimedia Tools and Applications*, vol. 77, no. 13, pp. 17069–17107, 2018.

[3] W.Nilmini. Manamperi, T.D. Abhayapala, J.A. Zhang, and P. Samarasinghe, "Drone audition: Sound source localization using on-board microphones," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022.

[4] "The mystery of the gatwick drone," `https://www.theguardian.com/uk-news/2020/dec/01/the-mystery-of-the-gatwick-drone`, Accessed: 2022-03-20.

[5] D. Barchiesi, D. Giannoulis, D. Stowell, and M.D. Plumbley, "Acoustic scene classification: Classifying environments from the sounds they produce," *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 16–34, 2015.

[6] A. Mesaros, T. Heittola, and T. Virtanen, "Tut database for acoustic scene classification and sound event detection," in *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 1128–1132.

[7] S. Safavi, T. Iqbal, W. Wang, P. Coleman, and M.D. Plumbley, "Open-window: A sound event dataset for window state detection and recognition," in *Proc. of International Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE'20). Tokyo, Japan*, 2020.

[8] Xinyu Fu, Eugene Ch'ng, Uwe Aickelin, and Simon See, "Crnn: a joint neural network for redundancy detection," in *2017 IEEE international conference on smart computing (SMARTCOMP)*. IEEE, 2017, pp. 1–8.

[9] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al., "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[10] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra, "Weight uncertainty in neural network," in *International conference on machine learning*. PMLR, 2015, pp. 1613–1622.

[11] Yarin Gal and Zoubin Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.

[12] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural information processing systems*, vol. 30, 2017.

[13] Alex Kendall and Yarin Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," *Advances in neural information processing systems*, vol. 30, 2017.

[14] G.A. Korn and T.M. Korn, *Mathematical handbook for scientists and engineers: definitions, theorems, and formulas for reference and review*, Courier Corporation, 2000.