Drone localization and identification using an acoustic array and supervised learning

Valentin Baron, Simon Bouley, Matthieu Muschinowski, Jerome Mars, Barbara Nicolas

▶ To cite this version:

Drone localization and identification using an acoustic array and supervised learning

Valentin Baron[a,b,c], Simon Bouley[a], Matthieu Muschinowski[c], Jérôme Mars[c], and Barbara Nicolas[b]

[a]MicrodB, 28 Chemin du Petit Bois, 69130 Ecully, France
[b]Univ Lyon, INSA-Lyon, Université Claude Bernard Lyon 1, UJM Saint-Etienne, CNRS, Inserm, CREATIS, UMR 5220, U1206, F-69100, Lyon, France
[c]Univ. Grenoble Alpes, CNRS, Grenoble-INP*, GIPSA-Lab, 38000 Grenoble, France

ABSTRACT

Drones are well-known threats both in military and civil environments. Identifying them accurately and localizing their trajectory is an issue that more and more methods are trying to solve. Several modalities can be used to make it such as radar, optics, radio-frequency communications and acoustics. Nevertheless radar suffers from a lack of reflected signal for small targets, optical techniques can be very difficult to set in natural environments with small targets, and self-flying drones can avoid radio detection. Consequently, this paper deals with the remaining acoustic modality and aims to localize an acoustic source, then to identify it as a *drone* or a *noise* using array measurements and a supervised learning method. The acoustic array allows to determine the source direction of arrival and a spatial filtering is performed to improve the signal to noise ratio. A focused signal is then obtained and used for characterizing the source. The performances obtained to identify this source as a *drone* or not are compared for two different learning models. The first one uses two classes *drone* and *noise* with a classic Support Vector Machine model while the second one is based on an One Class Support Vector Machine algorithm where only the *drone* class is learned. A database is generated with 7001 observations of *drone* flights and 3818 observations of *noise* recordings within a controlled environment where signals are played one at a time, given that an observation is a sequence of 0.2 s of signal. Results of localization show an average error concerning the elevation angle bounded to 3.7° whereas identification results on this database give 99.5 % and 95.6 % accuracies for the two classes approach and the one class approach, respectively. It is shown that this high accuracy is reached thanks to the intrinsic separability of the created data obtained by the different features that have been chosen to compute.

Keywords: Drone identification, Machine Learning, Acoustic array processing, Support Vector Machine.

1. INTRODUCTION

Drones are more and more used for both military and civil applications. Although they can help in most cases, they can also represent threats for valuable assets in these two fields. To monitor the potential misuse of such tools, it becomes essential to develop identification devices to detect, track and potentially neutralize them as quickly as possible. Radar-based detection tools are developed for this purpose but they rely on an active system that can be deceived if the reflected parts of the target are not enough present.[1] Radio frequency communications interception between the drone and its pilot can also be used for detection but models that do not require any control are already developed.[2] Video-based frameworks are also proposed but they are strongly dependent of the weather conditions[3] and the detection of flying objects is more complex than the well-developed detection schemes for autonomous cars.[4] The final modality studied within this contribution, is to work with the acoustics signals emitted by the drone, as it is obvious that a powerful drone is inherently noisy.[5] Teams already work on that subject using one microphone and supervised machine learning techniques for detection,[6] developing

---

acoustic array methods to track the target drones in time[7–9] or combining both aspects to track and identify them at the same time.[10] In light of the numerous different solutions proposed to tackle this problem, it is clear that the state of the art method is not defined yet. This work is trying to establish a rigorous framework that embed the promising solutions all together for both localization and identification of drones. It starts with a localization step using an acoustic array. A spatial filter is applied to signals toward the direction given from this localization, allowing an enhancement of the signal to noise ratio. This so-called focalization step ends up with an one dimensional audio signal to identify. This identification is carried out by machine learning thanks to two different approaches: a supervised model that differentiates *drone* signals from *noise* ones in a binary classification task and a semi-supervised model learned over *drone* signals that detects *noises* as outliers. In this study both approaches are considered using Support Vector Machine algorithms, the classical version for the binary problem (SVM) and the One Class extension for the outlier detection case (OC-SVM). These methods are validated against experimental data created in a controlled environment. The ground truth localization is assessed by a infrared positioning system and both *drone* and *noise* data are recorded using a 81-channels acoustic array. The contribution is organized as follows: the section 2 explains the methodology developed to build the whole framework. Section 3 describes the available data produced to test it. The localization and identification results are given and discussed in section 4. The conclusion in section 5 ends this study.

2. METHODOLOGY

The localization and identification process is given in Fig. 1 with different blocks that describe its steps. The localization allows to find the direction of arrival of a targeted sound source present near the acoustic array. The focalization produces an single audio signal from the array recordings that is filtered in the direction of arrival. Finally the identification classifies this sound signal as a *drone* or a *noise* comparing it to the previously learned cases. This contribution deals with the single acoustic source case, given that this source can be a *drone* or any different sound that will be called *noise* in the following.
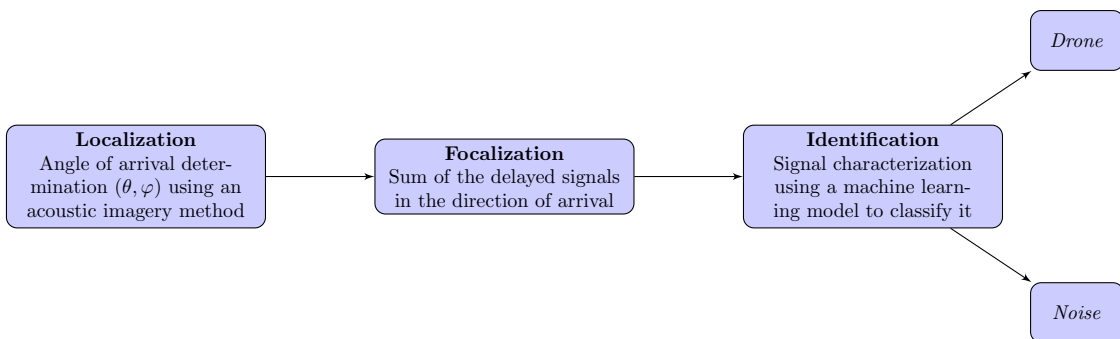

Figure 1. Sketch of the whole localization and identification of a drone. It is composed of three successive steps: the localization, the focalization and the identification. These steps enable the system to both track and identify an acoustic source as a *drone* or a *noise*.

2.1 Localization

The process starts with a localization step that aims to find the angular position of the acoustic source located around the array. The signals recorded by the acoustic array are cut into 0.2 s-long sequences, with a 50 % overlap, as depicted in Fig. 2. A cross spectral matrix Γ_k is computed for each sequence,[11] ordered by the index k. This data matrix gathers the information recorded by the microphones and specifically the delays between the channels, required to find the angle of arrival of the source. This is carried out with the help of a conventional beamforming (CBF, Eq. 1),

$$CBF_k(\theta, \varphi) = \frac{\mathbf{g}^*(\theta, \varphi)\ \Gamma_k\ \mathbf{g}(\theta, \varphi)}{Tr(\Gamma_k)\ ||\mathbf{g}(\theta, \varphi)||^2}\ , \tag{1}$$
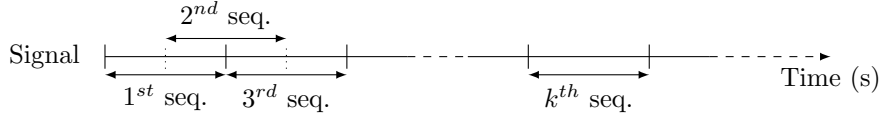
Figure 2. Cutting procedure of the recorded signals from the acoustic array.

with θ and φ the considered elevation and azimuth angles, $\mathbf{g}$ the steering vector of the array in this angular direction (θ, φ), $Tr(.)$ the trace operator and $.^*$ the conjugate transpose operator. It gives a normalized acoustic map[12] for every considered pair of angle and the maximum of the produced map is selected as the direction of arrival of the source, given by $(\theta_{m_k}, \varphi_{m_k})$.

2.2 Focalization

The goal of this step is to give as an output a spatially filtered signal in the direction of arrival of the sound source. The distances between the microphones and the first one taken as a reference are first projected on the vector that gives the direction of arrival $u_{m_k} = [\cos(\theta_{m_k})\cos(\varphi_{m_k}), \cos(\theta_{m_k})\sin(\varphi_{m_k}), \sin(\theta_{m_k})]^T$. Delays are recovered by dividing these distances by the speed of sound in air c_0. They are applied on every time signal that are then summed in order to give the final filtered signal for a given sequence k.

Finally the overlapped sequences are combined together by first applying a raised cosine window to ensure 0-amplitude samples on the borders of each sequence. Then they are summed according to their original position within the signal as described in Fig. 2, thanks to the overlap-add technique.[13]

2.3 Identification

This final step aims at classifying the localized and filtered source as a *drone* or a *noise*. Two ways are considered in this work: a supervised Support Vector Machine (SVM) model and a semi-supervised One Class Support Vector Machine (OC-SVM) one. They both work the 0.2 s sequences of focused signals. Consequently a so-called observation (Obs.) is a 0.2 s-long sequence of focused signal on which features detailed in section 2.3.3 are computed. To compare both models the accuracy is chosen and is given by the ratio between the number of correctly classified observations over the total number of observations.

2.3.1 Support Vector Machine

The first SVM model[14] is a supervised machine learning technique that classically works with two classes. It means that a first learning step is required, in which examples of both classes are given in order to build a boundary between them. This boundary is defined by the hyperplan that separates the two classes with a maximum margin. For this work the Scikit-learn algorithm version is implemented,[15] with no kernel and a coefficient C set to 1.

2.3.2 One Class Support Vector Machine

The second model is OC-SVM[16] which aims at detecting outliers from a cluster of observations. In a clustering scheme, outliers are defined as points that do not belong to any perceived cluster. The cluster to be identified here, composed of *drone* observations, is learned from example data to build an hyperplan between them and the origin of the considered vector space. Then new observations are classified according to their position compared to this hyperplan as members of the group or outliers. Although the method can use take advantage of kernels to go into higher dimension vector spaces, kernels are not required in this application because of the intrinsic separability of the used data, illustrated in section 4. The method relies on the parameter ν to build its hyperplan between the data and the origin of the vector space. It allows to tune the fraction of outliers within the learning observation set provided to learn. The lowest ν, the lowest fraction of outliers is authorized so almost all the data are taken to learn. Consequently ν represents directly the fraction of missed *drone* identifications. For the aimed application it is clear that a low ν is required because the learning set is composed only of *drone* examples.

Although OC-SVM is presented as an unsupervised model, in this study it is considered as a semi-supervised one since a first learning step is performed. So knowledge about the *drone* data that are given to build the hyperplan renders the process non fully unsupervised.

2.3.3 Features

32 features are computed[17] over three different domains: the temporal, spectral and cepstral domains. It leads to 96 features that describe the most exhaustively possible the input signals by means of statistical oriented, entropy oriented or shape oriented descriptors. It is shown that the use of this entire set of features allows to improve the classification performances.[18] Observations are separated in two groups, the learning and test observations, allowing to test the built boundaries over new observations. In order to set the ν parameter of the OC-SVM, the noise class learning observations are used: for different ν, their classification error is computed and the optimal ν is taken as the lowest before the increasing of the error, observed in Fig. 5, section 4.2. This process to set ν still supports the fact that the OC-SVM model is only semi-supervised.

3. EXPERIMENTAL DATASET

Data have been collected by a Siemens Sound Camera composed of 81 MEMS microphone channels that has already been used in other acoustic applications.[19] Three different drones have flown to create the database. Their position along time have been recorded by 12 Vicon T40S cameras.[20] The acoustic array and the three drones are shown in Fig. 3.
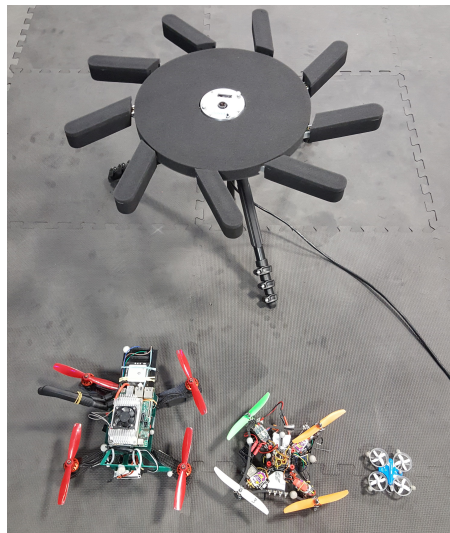

Figure 3. Used material for the database creation. On the top the Siemens Sound Camera acoustic array. On the bottom the three drones that have flown to create the database.

By the use of three different drones the class *drone* is ensured to encompass more different models. Indeed they have not the same motor technology, nor blade length, provoking different fundamental radiating frequencies. They have flown according to two numerically controlled trajectories of about one minute long with a distance to the array varying between 50 cm to 1.1 m. The ground truth is provided by the Vicon cameras.

The *noise* class is composed of 8 different sounds of around 30 s each: sinus signals at 1 and 3 kHz, white noise, sweep signal from 0 to 12 kHz, drill, engine, and helicopter noises. They are synthetically created or they come from a BBC sound database.[21] In practice they are played by a loudspeaker located at three different positions.

Table 1 sums up the number of usable flights (trajectory 1 and 2) provided by the acquired data with the drones, the number of different noises and positions taken by the loudspeaker, and the final number of observations that are effectively taken for learning (Lear.) and test (Test) steps. For the drones, the learning is

performed on trajectory 1 and the test on the trajectory 2, whereas only the helicopter noises are isolated for the test step (for the three positions of loudspeaker), and all the other noises are used for learning.

Table 1. Acquired data for *drone* and *noise* classes.

	Drone	*Noise*
# trajectory 1 / 2	15 / 11	
# noises / # positions		8 / 3
# Obs. (Lear. / Test)	3924 / 3077	2865 / 953

The two classes are slightly unbalanced but they still have a reasonable number of observations each to be able to describe themselves accurately. It is important to consider that the test step is performed on helicopter noises that seem to be the closest noises to the drone signals and they are not used at all during the learning step.

4. RESULTS

4.1 Localization

Localization results are illustrated using one flight of trajectory 1. All the flights have been processed and results are similar. Fig. 4a represents a comparison between the angular CBF tracking of the drone and the ground truth given by the Vicon cameras. A quantitative comparison of the elevation angle θ along this same trajectory is given in Fig. 4b.
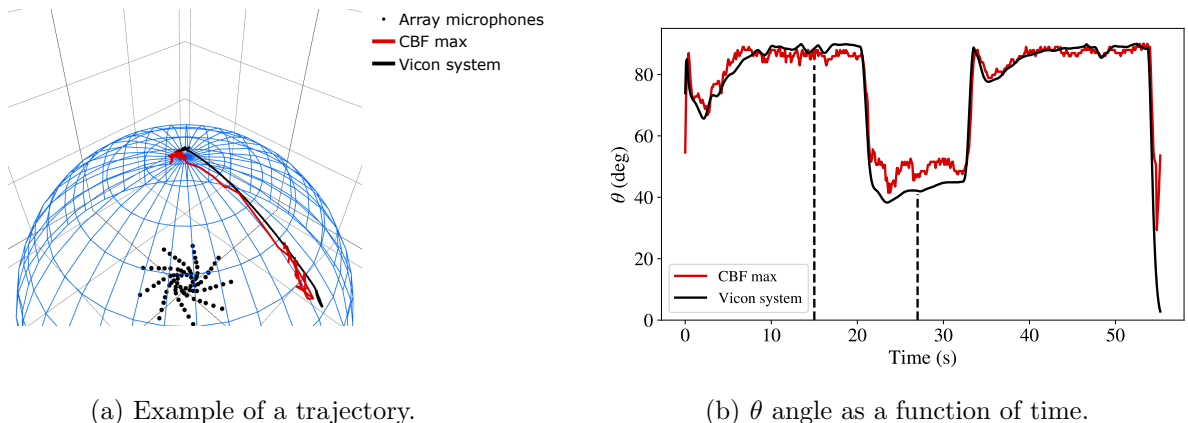

(a) Example of a trajectory.

(b) θ angle as a function of time.

Figure 4. Localization results with ▬ the position of the drone given by the Vicon system and ▬ the position of the drone given by the maximum of the conventional beamforming result. (a): illustration of a part of the trajectory 1, the array is located at the center of the meshed sphere. (b): elevation angle as a function of time for the trajectory 1. The trajectory part represented on subfigure (a) is given by the dashed vertical lines.

Acoustic localization given by the CBF maximum and ground truth localization given by the Vicon system are similar along the trajectory. However it is clear that for low elevation angles, a shift between them is observed with an acoustic localization that overestimates θ. Over the total trajectory, a mean error of $3.7°$ is made with the CBF estimate of the θ direction of arrival. The azimuth angle is not studied here because of excessive variations when the drone fly over the vertical of the array, which renders the analysis unstable. The illustration in Fig. 4a shows that its estimation is still accurate.

New data have been generated to investigate the causes of the elevation angle error and they are currently studied. Hypotheses are focused on potential room reflexions, CBF wrong estimation of the direction of arrival and diffraction on the array or presence of a directivity pattern with it.

4.2 Identification

Classification errors on drone and noise classes observations for the learning and the test groups are first compared as a function of ν in Fig. 5.
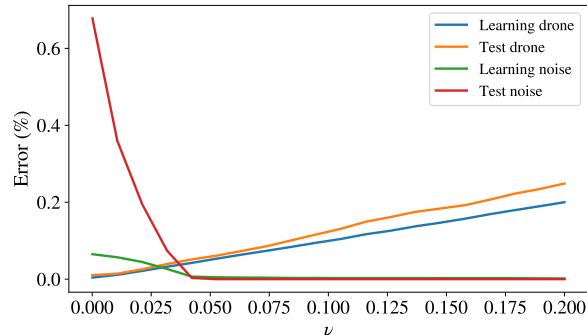

Figure 5. Classification error for different sets of observations as a function of the OC-SVM parameter ν. Classification errors for the *drone* class sets, curves ▬ and ▬, are varying linearly since ν directly controls the fraction of outliers present in the learning set. The *noise* class learning set, curve ▬, defines the parameter ν value at 0.05. Below this value this error is increasing. It must be noted that the model only learns on the *drone* class learning set. All the curves are produced once the hyperplan is built, looking at the position of the data compared to it.

The classification errors associated with both learning and test sets for the *drone* class vary linearly as a function of ν as expected given that ν directly controls the fraction of outliers allowed. For the *noise* class sets, the errors are null for values above $\nu = 0.05$ and they raise below this threshold. Indeed, when ν is set too low, the hyperplan is searched to take almost all the studied vector space into account. As a consequence even the *noise* class observations fall into the *drone* class.

So in the following, ν is set to 0.05 and both SVM and OC-SVM classify the observations of the test sets. The classification results are gathered in Tab. 2.

Table 2. Confusion matrices. (a): SVM classification results. (b): OC-SVM classification results.

(a) SVM		Predicted class	
		Drone	*Noise*
Reference class	*Drone*	99.4 %	0.6 %
	Noise	0 %	100 %

(b) OC-SVM		Predicted class	
		Drone	*Noise*
Reference class	*Drone*	94.2 %	5.8 %
	Noise	0 %	100 %

For the *noise* class, the classification results are the same: all the noises are classified in their class or in the drone one. However, some of the *drone* observations are classified as *noise* and these wrong classifications happen more frequently for the OC-SVM model. It follows that SVM has a 99.5 % accuracy when OC-SVM achieves 95.6 % accuracy for this classification task. Nevertheless even in the worst case using the OC-SVM model the implemented system classifies, over 10 consecutive seconds of drone signal, 9.42 s in the *drone* class against 0.58 s in the *noise* class. It may allow to build a strong identification system given that the wrongly classified observations are not necessarily consecutive within the signal.

An insight of why the two methods can give similar results is given by representing the histogram of the values taken by the feature that separate the best the two considered classes, here feature number 53, which encapsulate entropy information within the spectral domain. The histogram counts the number of observations that take a given value for the considered feature. These values computed over the learning sets are given in Fig. 6a whereas the test sets values are in Fig. 6b.

In the learning case, *drone* and *noise* class distributions are almost entirely separable thanks to this feature. As a consequence the binary problem solved by the SVM method can achieve accurate classification results.
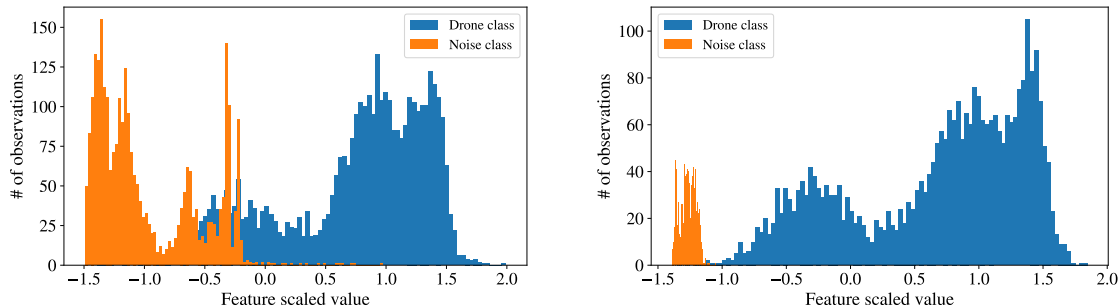
(a) Learning case histogram of feature 53.　　(b) Test case histogram of feature 53.

Figure 6. Histogram of the most significative feature over the drone and noise observation sets.

Then it can be seen that the *noise* class distribution is positioned within the negative values. This is why even without using a kernel (and consequently staying in the vector space of the computed features) the OC-SVM method can find a hyperplan that separates the *drone* and *noise* classes because it is equivalent to separate the drone class from the origin of the vector space.

When going into the test case, the *noise* class distribution is entirely positioned within negative values. If one has to set a threshold between the two distributions in the train case, it is clear that the test values of this feature for the *noise* case would be below this threshold. Consequently the 0 %-error obtained for this class is understandable. Inversely for the *drone* class, values would stay under this threshold giving the badly classified observations observed in Tab. 2.

Finally concerning the SVM against OC-SVM results, it can be seen that the best compromise to separate both classes in the train histogram should be to take a threshold slightly negative. However OC-SVM separates only compared to the origin and as a consequence it keeps more *drone* observations within the *noise* class than SVM can do as it is observed in Tab.2.

All these explanations highlight the fact that the computed features are well suited for the input acoustic signals they deal with. They efficiently separate them allowing the classification method to work easily afterward. However it has to be kept in mind that these data are academic in the sense of they deal with a single source case in a quiet environment so this analysis has to be continued for more complex configurations.

5. CONCLUSION

A framework to perform drone localization and identification is described. The localization step relies on an acoustic array method, the conventional beamforming, to find the angular direction of arrival of a nearby acoustic source. Then using the multiple available channels of the array a spatial focalization is made in this direction to obtain a filtered output that can be used for the next identification step. Two different machine learning models known as Support Vector Machine and One Class Support Vector Machine are implemented to identify the two defined classes *drone* and *noise*. The former uses a binary classification scheme whereas the latter only learns with *drone* class observations and *noise* class observations are identified as outliers. Data recorded with unique source configurations validate this framework with a localization error bounded under 3.7° on average. Identification results yield accuracies of 99.5 % and 95.6 % for the Support Vector Machine and the One Class Support Vector Machine models, respectively. These high accuracies are discussed looking at the values of the feature that best separate the two classes. With this first feature alone, a separation between the *drone* and the *noise* classes is clearly visible and it can be set more precisely using a binary classification that allows a negative threshold compared to a one class scheme.

Further work concerns the study of the deviation in localization for low elevation angle before starting to look at more complex configurations like multi-source or lower signal to noise ratio cases to finish the validation in an outside environment where various issues can happen at once.

ACKNOWLEDGMENTS

REFERENCES

[1] D. Shin, D. Jung, D. Kim, J. Ham, and S. Park, "A Distributed FMCW Radar System Based on Fiber-Optic Links for Small Drone Detection," *IEEE Transactions on Instrumentation and Measurement* **66**, 340–347 (Feb. 2017).

[2] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, "Vision-Based Odometry and SLAM for Medium and High Altitude Flying UAVs," *Journal of Intelligent and Robotic Systems* **54**, 137–161 (Mar. 2009).

[3] T. Multerer, A. Ganis, U. Prechtel, E. Miralles, A. Meusling, J. Mietzner, M. Vossiek, M. Loghi, and V. Ziegler, "Low-cost jamming system against small drones using a 3D MIMO radar based tracking," in [*2017 European Radar Conference (EURAD)*], 299–302 (Oct. 2017).

[4] A. Rozantsev, V. Lepetit, and P. Fua, "Detecting Flying Objects Using a Single Moving Camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**, 879–892 (May 2017).

[5] O. Gur and A. Rosen, "Design of Quiet Propeller for an Electric Mini Unmanned Air Vehicle," *Journal of Propulsion and Power* **25**, 717–728 (May 2009).

[6] A. Bernardini, F. Mangiatordi, E. Pallotti, and L. Capodiferro, "Drone detection by acoustic signature identification," *IS&T International Symposium on Electronic Imaging* **2017**, 60–64 (Jan. 2017).

[7] X. Chang, C. Yang, J. Wu, X. Shi, and Z. Shi, "A Surveillance System for Drone Localization and Tracking Using Acoustic Arrays," in [*2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*], 573–577 (July 2018).

[8] J. Tong, W. Xie, Y. Hu, M. Bao, X. Li, and W. He, "Estimation of low-altitude moving target trajectory using single acoustic array," *The Journal of the Acoustical Society of America* **139**, 1848–1858 (Apr. 2016).

[9] J. Busset, F. Perrodin, P. Wellig, B. Ott, K. Heutschi, T. Rhl, and T. Nussbaumer, "Detection and tracking of drones using advanced acoustic cameras," in [*Unmanned/Unattended Sensors and Sensor Networks XI; and Advanced Free-Space Optical Communication Techniques and Applications*], **9647**, 96470F, International Society for Optics and Photonics (Oct. 2015).

[10] A. Ramamonjy, E. Bavu, A. Garcia, and S. Hengy, "Source localization and identification with a compact array of digital MEMS microphones," in [*25th Congress on Sound and Vibration*], (July 2018).

[11] R. Merino-Martínez, P. Sijtsma, M. Snellen, T. Ahlefeldt, J. Antoni, C. Bahr, D. Blacodon, D. Ernst, A. Finez, S. Funke, T. Geyer, S. Haxter, G. Herold, X. Huang, W. Humphreys, Q. Leclère, A. Malgoezar, U. Michel, T. Padois, A. Pereira, C. Picard, E. Sarradj, H. Siller, D. Simons, and C. Spehr, "A review of acoustic imaging methods using phased microphone arrays," *CEAS Aeronautical Journal* **10**, 197–230 (Mar. 2019).

[12] G. Elias, "Experimental techniques for source location," in [*Aeroacoustics and Active Noise Control, Lecture Series, Von Karman Institue for Fluid Dynamics*], (1997).

[13] J. Driedger and M. Müller, "A Review of Time-Scale Modification of Music Signals," *Applied Sciences* **6** (Feb. 2016).

[14] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning* **20**, 273–297 (Sept. 1995).

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research* **12**, 2825–2830 (2011).

[16] B. Schlkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R.Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation* **13**, 1443–1471 (July 2001).

[17] M. Malfante, J. Mars, and M. Dalla Mura, "Automatic analysis architecture." https://zenodo.org/record/1216028 (2019). Accessed: 26/02/2019.

[18] M. Malfante, M. Dalla Mura, J.P. Metaxian, J. Mars, O. Macedo, and A. Inza, "Machine Learning for Volcano-Seismic Signals: Challenges and Perspectives," *IEEE Signal Processing Magazine* **35**, 20–30 (Mar. 2018).

[19] J. Antoni, T. Le Magueresse, Q. Leclère, and P. Simard, "Sparse acoustical holography from iterated Bayesian focusing," *Journal of Sound and Vibration* **446**, 289–325 (Apr. 2019).

[20] P. Merriaux, Y. Dupuis, R. Boutteau, P. Vasseur, and X. Savatier, "A Study of Vicon System Positioning Performance," *Sensors* **17**, 1591 (July 2017).

[21] BBC, "bbc.co.uk - ©copyright [2019] bbc." http://bbcsfx.acropolis.org.uk/ (2019). Accessed: 22/02/2019.