

Acoustic classification of individual cat vocalizations in evolving environments

Stavros Ntalampiras
University of Milan
Milan, Italy
stavros.ntalampiras@unimi.it

Danylo Kosmin and Javier Sanchez
Akvelon Inc.
Washington, USA
danylo.kosmin@akvelon.com, javier@meowtalk.app

Abstract—This paper is focused on the classification of vocalizations characterizing intents of individual cats. Cats vocalize in order to convey different emotions and/or intents and although their repertoire/vocabulary may not be universal, it exhibits consistent characteristics on an individual basis. In this work, we present a complete pipeline for processing streams of audio, with the twofold goal being both detection as well as interpretation of cat vocalizations. The proposed system is based on YAMNet pre-trained deep network where we apply meaningful modifications addressing the requirements of the task-at-hand. Interestingly, the overall system is able to run in real-time on modern smartphones using the developed application. At the same time, we address the non-stationarity problem meaning that class dictionary is updated on-the-fly following user recommendations. After extensive experiments, we show that the proposed pipeline achieves quite satisfactory detection and recognition accuracy. To the best of our knowledge, this is the first attempt in the related literature to address continuous detection and interpretation of cat vocalizations in a real-time fashion.

Index Terms—Bioacoustics, cat vocalizations, deep learning, transfer learning, learning in non-stationary environments.

I. INTRODUCTION

The scientific field of bioacoustics is receiving increased attention over the past decades as it addresses a variety of applications facilitating diverse scopes including biological research and education of young scientists [1]–[3]. The typical pipeline of such systems encompasses two modules of fundamental importance, i.e. *signal processing* and *pattern recognition*. The former aims at the design of mechanisms able to capture and reveal distinctive information characterizing the problem-at-hand. The latter includes modeling the extracted parameters using machine learning technologies allowing the automatic characterization of novel audio patterns [4]–[6].

Animal vocalizations comprise a very efficient way of animal-animal and animal-human communication since the specific modality a) does not require visual contact between emitter and receiver, b) can travel over long distances, and c) can carry the information content under limited visibility conditions such as dense forests, night, rain etc. [7], [8]. Animals produce a variety of sounds in order to communicate for reasons that may be vital to their existence [1]. Vocalizations are employed for mate attraction, territorial defence and for early warning of other members of the species regarding the presence of dangerous predator. These sounds range from short simple calls (also called pulses, syllables, or notes) to versatile long songs, which are composed of a complex hierarchy of syllables (very common in singing birds). The communication strategy and the diversity of the frequency structure of the sound pattern depend heavily on the environmental context of the living organism.

Cats a) are one of the most common companion animals, b) spend a considerable amount of their life with humans and c) more often than not, are recognised as social partners by their owners [9], [10]. Unfortunately, there is not a lot of works regarding automatic processing of cat vocalizations assisting the understanding of their structure

as well as the principles regulating cat-human communication and interaction [11]. There are few works based on the visual modality [12] but the acoustic modality is largely unexplored, thus this work is focused on the automatic understanding of cat vocalizations. The closest paper to this work is presented in [9] where an audio pattern recognition system is designed based on Mel-frequency cepstral coefficients and temporal modulation features combined with a graph-based classifier composed of hidden Markov models. Three classes of cat vocalizations are considered, i.e. waiting for food, isolation, and brushing [13]. It emerges that there is not a standardized feature set, while the case of non-stationary environments is not considered. Indeed, it might be the case that new classes of cat vocalizations might appear during system operation which need to be considered by the mechanism responsible for their automatic processing and characterization. The literature does not include other methods for classifying cat vocalizations able to be executed on a standard smart phone device, let alone addressing the case of evolving environments.

This work proposes an algorithm for processing raw streams of audio potentially including cat vocalizations. We present a deep learning framework addressing the automatic recognition of cat vocalizations. Motivated by the gaps existing in the related literature, we present a solution that

- carries out both detection and interpretation of cat vocalizations,
- operates in unrestricted open-end environments,
- elaborates on a standardized feature set,
- incorporates non-stationarities on-the-fly based on user’s input,
- customizes interpretation models during operation learning properties at the individual level, i.e. creating personalized models, and
- encompasses the entire framework in a suitably-developed mobile application able to consider potential input(s) received by its user.

We consider a large dictionary of classes including the following: *angry, defensive, fighting, hunting, happy, mating, mother call, pain, rest*. After extensive experiments, we present results systematically assessing the performance of the proposed system.

II. PROBLEM FORMULATION

In classifying cat vocalizations, we assume availability of a training set T^s encompassing single-channel recordings annotated according to cat’s intent coming from dictionary \mathcal{D} . Composition and cardinality of \mathcal{D} are known only up to a certain extent, i.e. dictionary $\mathcal{D} = \{C_1, \dots, C_n\}$, where C_i denotes the i -th class, meaning that a-priori unknown classes may appear during system’s operation. Such classes are inserted by the user during system’s operation. At the same time, we assume that each class follows a consistent, yet unknown probability density function, which is typical for generalized audio processing systems [5].

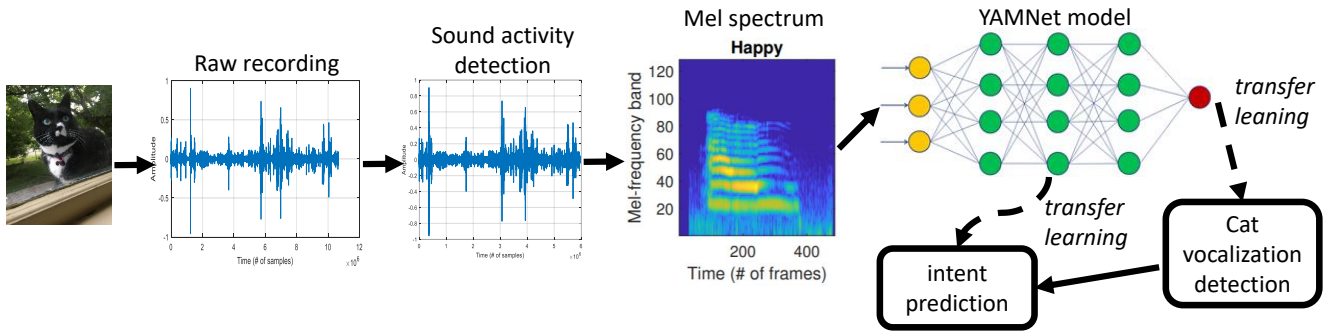


Fig. 1. The pipeline of the proposed framework encompassing a) acquisition of raw recordings, b) sound activity detection, c) extraction of Mel-spectrogram, d) cat vocalization detection and e) intent prediction using YAMNet model.

The overall goal of the system is to *detect* and *identify* an individual cat’s intent from the available audio signal, while *incorporating* new user-defined classes in an online fashion.

III. THE PROPOSED FRAMEWORK

This section describes the proposed pipeline enabling classification of individual cat vocalizations according to cat’s intent.

A. Feature extraction

Following the success of summarized spectrograms in the audio signal processing literature [14]–[16] including YAMNet model¹, we operate on Mel-scaled spectrograms extracted based on a triangular filterbank of 128 log-energies. To this end, the audio signal is cut in sliding windows on which short-time Fourier transform (STFT) is obtained. The STFT result is processed by the Mel-filterbank and finally the logarithm is computed. Interestingly, such a standardized feature extraction process practically eliminates the need to design acoustic features tailored to the problem at hand, which is typically a time-consuming and costly process since it heavily relies on domain expertise.

B. Model architecture

Given the fact that we are aiming at a mobile application, we employed YAMNet model which is based on the depthwise-separable convolution architecture [17]. YAMNet is a publicly-available pre-trained deep network able to predict 521 audio event classes according to the definition of the AudioSet-YouTube corpus [18]. YAMNet follows a VGGish architecture including 3.7M weights, i.e. 1/20th the size of VGGish (3.7M vs. 72.1M). At the same time, in terms of computational needs, YAMNet requires approximately 1/10th the resources of VGGish (69M vs. 864M).

Based on YAMNet model, we designed a structure suitable to address the present problem as shown in Fig. 2. More specifically, the network operates on input mel-spectrograms of size (32,48,32). Such inputs are processed by an 1-D convolutional layer, followed by a series of 2-D convolutional layers. The kernel size is 3×3 while stride and number of filters vary between 1-2 and 64-1024 respectively as demonstrated in Fig. 2. A global average pooling layer is included aiming at minimizing potential overfitting by reducing the total number of parameters of the model [19]. Finally, the network is complemented by two fully-connected layers of sizes 1024 and 64, along with a softmax one with dimensionality equal to the number of classes existing in \mathcal{D} . The last layer is responsible for the output of the network and the final prediction w.r.t. a given input. Importantly, only

the final fully-connected and softmax layers are in need of training using an appropriate dataset of cat vocalizations representative of the problem-at-hand.

C. Model customization and personalization to individual cats

Since YAMNet is not specifically trained to address the problem which is the target of this work, a suitable customization process was designed making it able to address classification of cat vocalizations. It relies on the representation extracted out of YAMNet as well as an additional layer responsible to carry out the classification and as such the number of its nodes is equal to the number of classes existing in \mathcal{D} . The motivation behind such a design lies in the fact that YAMNet’s representation captures discriminative characteristics of the input audio, which can be transferred to other problems. The specific approach belongs to the family of *transfer learning* techniques [20], [21], where only the last classification layers require training. As such, the model is based on the well-learned representation of audio structures provided by YAMNet, which is trained on a 2 million human-labeled 10-second samples, combined with a suitable classification mechanism able to address the task-at-hand.

More specifically, the present modeling scheme is based on transferring knowledge learnt by YAMNet facilitating our pipeline in two modeling stages:

- detection of cat vocalizations in continuous streams of audio deriving model $\mathcal{M}_{\mathcal{D}}$, and
- recognition of the included intent deriving model $\mathcal{M}_{\mathcal{R}}$.

As such, our set of models encompasses two feedforward neural networks. Interestingly, the initial part based on YAMNet is common and thus, is meant to run only once, accelerating the overall recognition process. Both $\mathcal{M}_{\mathcal{D}}$ and $\mathcal{M}_{\mathcal{R}}$ are updated using incoming user-generated data. $\mathcal{M}_{\mathcal{R}}$ is updated on a daily basis as long as the respective data becomes available, while $\mathcal{M}_{\mathcal{D}}$ is updated over longer time periods, e.g. monthly, mainly because of its large size, so that there is a reasonably-sized bandwidth requested by the user.

D. Algorithm to update class dictionary based on user’s input

Following the general pipeline shown in Fig. 1, the system processes incoming cat vocalizations and assigns them a label existing in \mathcal{D} . This label is presented to the user via the mobile application and in case he/she does not agree with it, he/she is able to both indicate it and alter it via the corresponding functionality. More specifically, there are two possibilities for the user:

- choosing another class existing in \mathcal{D} , thus the model structure depicted in Fig. 2 remains unaltered, or

¹<https://github.com/tensorflow/models/tree/master/research/audioset/yamnet>

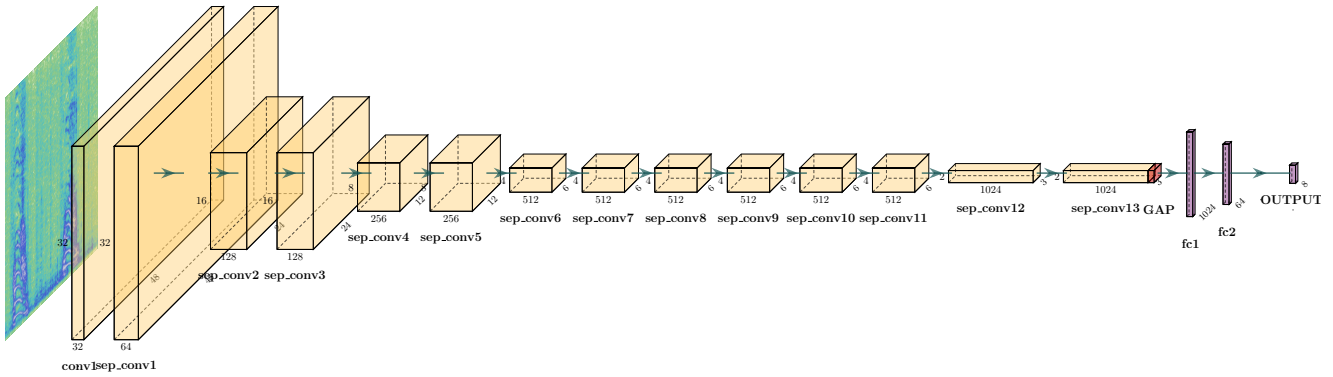


Fig. 2. The architecture of YAMNet model which is complemented by two fully-connected layers along with a softmax one with dimensionality equal to the number of classes.

a) the user creates a novel class previously unknown to \mathcal{D} . In that case, the system needs to incorporate the newly presented class and adapt accordingly.

The proposed algorithm belongs to *passive* concept drift detection class [22], [23] since it doesn't actively seek changes in the environment, unlike for example approaches such as [24]. It relies on user's input indicating the appearance of a new class via the mobile application. Such an indication increases the dimensionality of \mathcal{D} by 1 and the new class is populated by the newly captured data. This data is used to retrain the last layers of the network in Fig. 2 while output's dimensionality is increased by 1 as well. As more data becomes available, the specific layer learns more classes depending on user's input.

It should be highlighted that at this point the learned model is personalized to the individual cat and as such, it has been adapted to its characteristics. Thus, the updated model is going to be used only for this individual without affecting models adapted to the characteristics of others. This is also in line with the related literature stating that there is not a universal vocabulary/dictionary encompassing every class of cat vocalizations. Interestingly, such a functionality allows the user to create his/her own classes and develop a personalized unique interaction with the companion animal. This would be true as long as the data is annotated by the user consistently, i.e. in case of erroneous annotation, if the errors are consistent the cat's vocabulary will not be affected nor the model's ability to generalize over new recordings. Finally, learning both model customization and personalization to individual cats, i.e. the layers operating on top of YAMNet are trained based on Adam optimization algorithm [25].

IV. MOBILE APPLICATION

The entire pipeline, starting from data collection to cat vocalization detection and intent interpretation, has been suitably implemented in a mobile application called *MeowTalk*, available both on Google Play and Apple App Store. Keeping in mind the specifications of the present research and more specifically the ability to consider user-driven non-stationarities, we outline the considerations and requirements for the development of the mobile application towards ensuring the collection of accurate information. As such, the mobile application is seen as the extension of the current research with functionalities serving and facilitating its goals.

The application provides users with an easy way to detect and label their cat vocalizations. Out of the box, the application is loaded with a model that can detect general cat vocalizations along with basic

intents as defined in section I following previous research in the area [9].

The application has two recognition modes: *single* and *continuous* intent detection mode as show in Fig. 3 right and middle part respectively. In single intent detection mode the application will detect a single cat vocalization, submit it to the intent classifier, present the results to the user and ask the user to acknowledge the correct classification or to provide a correction. The correct classifications are used to train and improve the cat specific model for the user's cat. In continuous recognition mode the application performs cat vocalization and intent classification sequentially but does not offer the user the ability to correct or acknowledge the system's predictions.

Other features offered by the app include allowing the user to create multiple cat profiles (typically depending on the number of cats he/she owns), access their cat's profile regardless of the mobile device they use, provide custom labels and review the intent classification history along with playing back of the recorded sound as shown in Fig. 3 left part.

The MeowTalk Application is available both in Google Play and Apple App Store (Android and iOS versions)².

V. EXPERIMENTAL SET-UP AND RESULTS

This section describes the a) employed dataset, b) figures of merit, c) feature and model parameterization, and d) obtained results.

A. Dataset

The dataset includes the following classes: *angry, defense, fighting, happy, hunting, mating, mother call, pain, rest*. Data were taken from the worldwide web (YouTube, freesound, etc.) and are balanced across classes, while annotation follows the description in [26]. The dataset includes approximately 200 samples per class with an average duration of 1.7s. The sampling rate is 16kHz with 16 bit quantization.

B. Figures of merit

There are three aspects in need of evaluation of the proposed system: a) detection of cat vocalizations, b) intent recognition using dictionary \mathcal{D} , c) intent recognition using user-augmented dictionary \mathcal{D}' .

Regarding detection we used the following two metrics:

- *False positive index* (FP): it counts the times a cat vocalization is detected when there is not (percentage).

²Google Play <https://play.google.com/store/apps/details?id=com.akvelon.meowtalk>, Apple App Store <https://apps.apple.com/us/app/id1524038030>

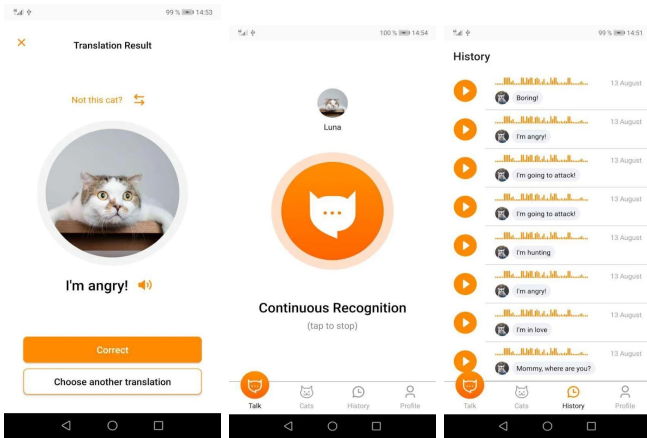


Fig. 3. MeowTalk application screen shots, recognition results screen, continuous recognition screen, history screen

- *False negative index* (FN): it counts the times a cat vocalization is not detected when there is (percentage).

Regarding intent recognition, we employed confusion matrices both in \mathcal{D} and \mathcal{D}' . In this way, we are able to thoroughly assess existing misclassifications and observe how the system is performing when a new class is inserted by the user.

C. Feature and model parameterization

Regarding feature extraction, we used windows of size 30ms moving by 10ms in each step to calculate the log-Mel spectrograms. The FFT size is 1024. Moving to the classification model, the receptive field sizes are given in Fig. 1. The activation function is softmax at learning rate is 0.0001 while the maximum number of permitted iterations is 50. Finally, the network is trained on minimizing categorical cross-entropy loss. For reporting the experimental results, we followed the ten-fold cross validation protocol and we present the averaged values.

D. Results

As of now (28/8/2020), there are 150 registered users out of which 101 users have created cat profiles with at least one recording. Overall, current availability is 3,000 samples with cat vocalizations, while the average duration is 1.5s.

During the first experimental phase, we assessed the cat vocalization detection model via FP and FN rates as defined in section V-B. The present experiment was conducted on real-world data captured by users via their mobile phones. As such, the reported rates do not concern any simulation case. On the contrary, we considered approximately 3,000 samples to compute FNs and the rest for FPs.

FP and FN rates are shown in Table I. There, we see that FPs are kept within low rates (5.7%) while FNs are considerably higher (9.1%). Interestingly, most false positives come from people mimicking a cat vocalization, which is actually a positive type of error as it boosts user engagement. At the same time, FNs are not so satisfactory mainly because of the variability of the non-cat vocalization class which not only includes a vast gamut of sound events but also many different environments. Our users are testing the mobile application in both indoor and outdoor environments without any type restriction; as such, acoustic conditions (e.g. reverberation, noise, etc.) vary substantially and present an uncontrolled obstacle of potentially increased level of interference. Another important source of mis-detections/classifications is the mismatch between the

TABLE I
FP AND FN RATES ASSESSING THE PERFORMANCE OF THE PROPOSED SYSTEM IN DETECTING CAT VOCALIZATIONS.

Approach	FP (%)	FN (%)
YAMNet-TL	5.7	9.1

recording equipment. However, at the moment, normalization both at time and frequency domains has minimized this issue so far.

Moving to intent recognition, we provide the results in the confusion matrix form in Fig. 4. The one corresponding to stationary conditions is shown on the left part while a non-stationary case of a user who enhanced \mathcal{D} by one class is on the right part. The average recognition rate under stationary \mathcal{D} is 90% demonstrating the reliability of the proposed model. The class recognized with the highest rate (100%) is *resting*, while classes *happy* and *pain* are the ones recognized with the poorest accuracies, i.e. 65% and 76% respectively. Most misclassifications concern errors made between these two classes. Another important source of misclassification concerns the pair of classes *defence-hunting*. Overall, we argue that the performance of the system is more than satisfactory, while most errors are due to the high intra-class variability of the dataset. To the best of our knowledge, there is no method in the literature able to learn cat vocalizations in evolving environments in order to carry out a comparison.

The right part of Fig. 4 tabulates the confusion matrix calculated when a specific user manually added an extra class, which is representative amount of current users. The added class is labeled as *come in* possibly denoting the case where the cat asks its owner to enter a specific area. In this case, the model is retrained as explained in section III-D so that the new class is incorporated in \mathcal{D} . Conveniently, retraining takes place on-the-fly as soon as a new class is presented to the system while the computational expensive part is carried out when the mobile phone is practically inactive (i.e. the user does not use it). In the non-stationary case the average recognition rate is 87.1%, slightly lower than the previous one. We observe that the newly added class is recognized with 86% accuracy which shows the efficacy of the presented approach in considering additional to \mathcal{D} classes. The specific evaluation process is not thorough and should be considered as a proof-of-concept, while extensive experiments will be presented in our future works.

In conclusion, we argue that the proposed pipeline is effective in every included module, i.e. detecting and recognizing cat vocalizations even when *a-priori* unknown classes need to be processed. Interestingly, such encouraging results are confirmed by user satisfaction surveys concerning the adoption of the mobile application.

VI. CONCLUSIONS

This article presented a complete pipeline starting from raw streams of audio, where individual cat vocalizations are detected and interpreted automatically with more than satisfactory performance. Interestingly, such functionality is able to be executed entirely on reasonably high-end mobile phones via the developed application. Transfer learning of knowledge learnt by YAMNet comprises the cornerstone of the proposed method. Importantly and unlike most machine learning based algorithms, the specific solution is able to operate in non-stationary environments and learn novel user-defined classes online.

Due to the automatized periodic model updating (both \mathcal{M}_D and \mathcal{M}_R), overall performance is expected to improve over time. Slight boost was already observed in the initial weeks of the mobile

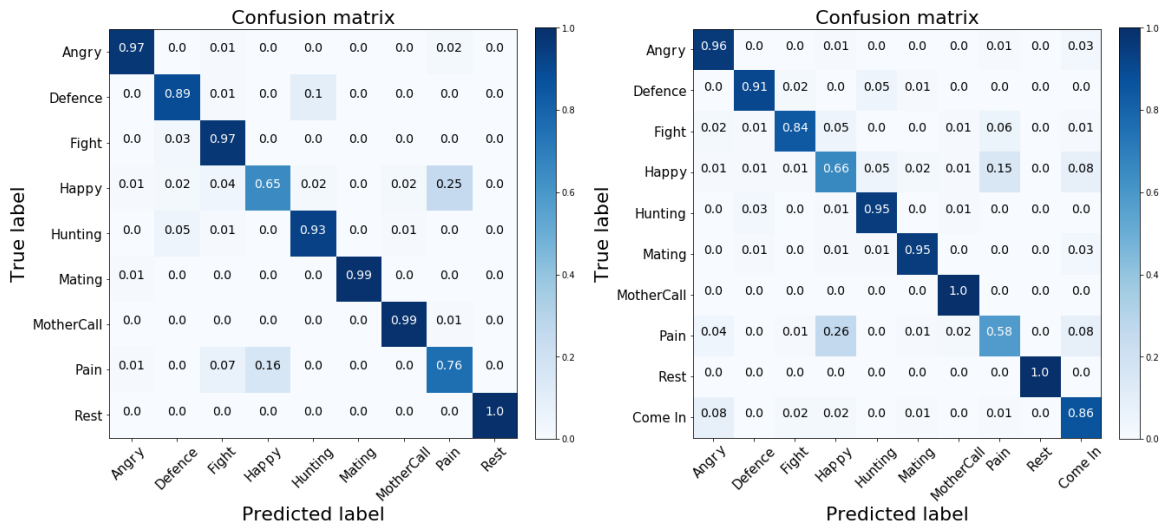


Fig. 4. Confusion matrices obtained by the proposed model w.r.t stationary (left) and non-stationary (right) class dictionary \mathcal{D} .

application usage. Nonetheless, there are several ways worth exploring towards improving the current pipeline: 1) explore diverse audio representations in parallel, e.g. features based on constant-q transform, 2) data augmentation, especially when learning in non-stationary data-limited environments, 3) incorporation of contextual non-acoustic information (routine, sex, castration, species, etc.). Finally, a fruitful direction would be to figure out ways to efficiently manage the constantly growing amount of data and users. This might assist in exploiting similarities and/or dissimilarities at the user and/or individual cat level, e.g. by sharing model parameters/layers, suitable clustering techniques, etc.

REFERENCES

- [1] Paola Laiolo, "The emerging significance of bioacoustics in animal species conservation," *Biological Conservation*, vol. 143, no. 7, pp. 1635–1645, July 2010.
- [2] D. Stowell, E. Benetos, and L. F. Gill, "On-bird sound recordings: Automatic acoustic recognition of activities and contexts," *IEEE/ACM TASLP*, vol. 25, no. 6, pp. 1193–1206, June 2017.
- [3] Iraklis Rigakis, Ilyas Potamitis, Nicolaos-Alexandros Tatlas, Ioannis Livadaras, and Stavros Ntalampiras, "A multispectral backscattered light recorder of insects' wingbeats," *Electronics*, vol. 8, no. 3, Mar. 2019.
- [4] M. A. Raghuram, Nikhil R. Chavan, Ravikiran Belur, and Shashidhar G. Koolagudi, "Bird classification based on their sound patterns," *Intern. Journal of Speech Tech.*, vol. 19, no. 4, pp. 791–804, Sept. 2016.
- [5] Stavros Ntalampiras, "Generalized sound recognition in reverberant environments," *Journal of the Audio Engineering Society*, vol. 67, no. 10, pp. 772–781, Oct. 2019.
- [6] P. Rai, V. Golchha, A. Srivastava, G. Vyas, and S. Mishra, "An automatic classification of bird species using audio feature extraction and support vector machines," in *ICICT*, Aug 2016, vol. 1, pp. 1–5.
- [7] Klaus Riede, "Monitoring biodiversity: Analysis of amazonian rainforest sounds," *Ambio*, vol. 22, no. 8, pp. 546–548, 1993.
- [8] Andrew Farnsworth, Daniel Sheldon, Jeffrey Geevarghese, Jed Irvine, Benjamin Van Doren, Kevin Webb, Thomas G. Dietterich, and Steve Kelling, "Reconstructing velocities of migrating birds from weather radar – a case study in computational sustainability," *AI Magazine*, vol. 35, no. 2, pp. 31, June 2014.
- [9] Luca A. Ntalampiras, Stavros Ludovico, Giorgio Presti, Emanuela Prato Prato Previde, Monica Battini, Simona Cannas, Clara Palestrini, and Silvana Mattiello, "Automatic classification of cat vocalizations emitted in different contexts," *Animals*, vol. 9, no. 8, pp. 543, Aug. 2019.
- [10] Matilda Eriksson, Linda J. Keeling, and Therese Rehn, "Cats and owners interact more with each other after a longer duration of separation," *PLOS ONE*, vol. 12, no. 10, pp. e0185599, Oct. 2017.
- [11] Yagya Raj Pandeya, Dongwhoon Kim, and Joonwhoan Lee, "Domestic cat sound classification using learned features from deep neural nets," *Applied Sciences*, vol. 8, no. 10, pp. 1949, Oct. 2018.
- [12] Jessica L. Owens, Mariana Olsen, Amy Fontaine, Christopher Kloth, Arik Kershenbaum, and Sara Waller, "Visual classification of feral cat felis silvestris catus vocalizations," *Current Zoology*, vol. 63, no. 3, pp. 331–339, Feb. 2017.
- [13] Luca A. Ludovico, Stavros Ntalampiras, Giorgio Presti, Simona Cannas, Monica Battini, and Silvana Mattiello, "Catmeows: A publicly-available dataset of cat vocalizations," in *MultiMedia Modeling*, 2021, pp. 230–243, Springer International Publishing.
- [14] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, May 2019.
- [15] S. Ntalampiras, "Deep learning of attitude in children's emotional speech," in *IEEE CIVEMSA*, 2020, pp. 1–5.
- [16] Yu Su, Ke Zhang, Jingyu Wang, and Kurosh Madani, "Environment sound classification using a two-stream CNN based on decision-level fusion," *Sensors*, vol. 19, no. 7, pp. 1733, Apr. 2019.
- [17] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017.
- [18] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.
- [19] Min Lin, Qiang Chen, and Shuicheng Yan, "Network in network," 2013, cite arxiv:1312.4400Comment: 10 pages, 4 figures, for iclr2014.
- [20] R. Ribani and M. Marengoni, "A survey of transfer learning for convolutional neural networks," in *SIBGRAP-T*, 2019, pp. 47–57.
- [21] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.
- [22] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, "Learning in non-stationary environments: A survey," *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 12–25, Nov 2015.
- [23] João Gama, Indrè Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–37, Apr. 2014.
- [24] S. Ntalampiras, "Automatic analysis of audiostreams in the concept drift environment," in *2016 IEEE MLSP*, Sept 2016, pp. 1–6.
- [25] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," 2014, 3rd International Conference for Learning Representations, San Diego, 2015.
- [26] Susanne Schotz, "The secret language of cats: How to understand your cat for a better, happier relationship," Tantor Audio; Unabridged edition, 2018.