Improving beehive state prediction from audio using relevant timbre features and deep neural networks

Leila Khellouf, Dominique Fourer
IBISC Lab (EA 4526)
University of Evry-Val-d'Essonne / Paris-Saclay, France,
Email: lkhellouf90@yahoo.com — dominique.fourer@univ-evry.fr

Dominique Cassou-Ribehart, Jean-Paul Gavini

Starling Partners
Paris, France
Email: dcr@starling.partners — jpg@starling.partners

Abstract

In recent years with the decrease of the honey bees population, interest for smart beekeeping is growing. Recent works propose new robust methods and low-cost systems for remotely monitoring the health state of a beehive from field audio recording. To this end, the future methods should obtain the best accuracy and reduce the computation cost and the amount of transmitted data. Hence, this paper proposes a comparative study of several supervised techniques applied on a publicly available dataset for predicting the presence or the absence of the queen. We also introduce a novel method based on audio timbre features whichs lead us to a significant reduction of the number of features coefficients and an improvement of the prediction robustness when compared to existing approaches based on mel frequency cepstral coefficients (MFCC) or short-time Fourier transform (STFT).

Key words: Beehive monitoring, audio timbre features, smart beekeeping, deep learning.

1 Introduction

Honey bee is one of the important pollinator insects for flowers, fruits and vegetables. Since decades, several regions of the world are suffering from a decrease of the bees population and of their economic activity based on the managed honey bee colonies. A large number of factors such as pests and diseases are being investigated [1, 2], but require accurate monitoring techniques to assist beekeepers. Bees produce specific sounds when exposed to stressors such as failing queens, predatory mites, and airborne toxicants, however experienced beekeepers are not always able to explain the exact causes of the sound changes without a hive inspection. Unfortunately, hive inspections disrupt the life cycle of bee colonies and put additional stress factors to the bees [3].

In order to develop systems for automatically discriminating different states of a beehive, several works propose to analyze the audio signature of beehive through a machine learning approach [4]. In [5, 6], the authors propose a method which combine the short-time Fourier transform of the analyzed audio with convolutional neural networks to discriminate bee sounds from the chirping of crickets and ambient noise. This approach outperforms classical machine learning methods such as k-nearest neighbors, support vector machines or random forests for classifying audio samples recorded by microphones deployed above landing pads of Langstroth beehives. The detection of the bee queen presence appears to be one of the most important tasks for smart beekeeping and is addressed [7] with a complete beehive machine-learning-based audio monitoring system. More recently, in [3, 8] the authors use their experience in audio signal processing and investigate the use of mel-frequency cepstral coefficients (MFCC), and spectral analysis of the sinusoidal components (or modes) as input features of a supervised classification method to automatically predict the absence or the presence of the queen in a beehive from the analyzed audio signal. The present paper pursue these works and propose to improve the bee queen presence prediction accuracy of the existing approaches with a specific attention to the number of the computed feature coefficients which should be as low as possible when an embedded system is used in a real-world application scenario.

This paper is organized as follow. In Section 2, we propose a supervised methodology for automatically predicting the health state of a honey bee colony from instantaneous field sound recording. In Section 3, we comparatively assess

a new proposed technique with several state-of-the-art methods with a consideration for the prediction accuracy and for the number of computed features. Finally, our results are discussed in Section 4 with eventual future works.

2 Proposed approach

In order to predict the health state of a beehive, we use the supervised learning approach illustrated in Fig. 1 which consists in (step-1) extracting features from the analyzed audio signals and then (step-2) computing labels using a classification method trained on annotated audio examples.

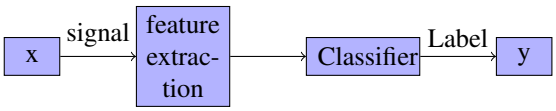


Fig. 1. Overview of the proposed approach.

2.1 Feature extraction

2.1.1 Mel-frequency cepstral coefficients

Mel-frequency cepstral coefficients (MFCC) is a classical representation based on the non linear mel frequency scale which finds a large number of audio applications such as speech recognition and music information retrieval [9,10]. They are introduced in several studies for beehive audio analysis in [11, 8] for which they provide promising results. Given an input signal x, the computation of the MFCCs follows the following steps:

1. application of a sliding window along x
2. computation of the Fourier transform $X(\omega)$
3. application of a mel filter bank over $|X(\omega)|$ to obtain $\mathcal{X}(k)$ with $k \in [1;K_m]$, K_m being an arbitrary defined number of coefficients
4. the instantaneous cepstral coefficients are given by the discrete cosine transform (type-2 is used this paper) of signal $\log(|\mathcal{X}(k)|^2)$.

In this study, we chose $K_m = 20$ as proposed in [8] and we used the python implementation provided in librosa [10].

2.1.2 Audio timbre features

Peeters *et al.* proposed in [12] a large set of audio features which convey information about the perceived timbre of a given sound (e.g. brightness that is commonly associated to the spectral centroid [13]). These audio features can be organized as follows (cf. table 1):

1. Temporal features describe the temporal evolution of a signal
2. Harmonic features are deduced from the estimated fundamental frequencies (F_0) and use the prior waveform model of quasi-harmonic sounds. Moreover, the tonal part of sounds can be isolated from signal mixture and be described (e.g. noisiness, inharmonicity, etc.).
3. Spectral features use the time-frequency representation of the signal (i.e. STFT) without prior waveform model (e.g. spectral centroid, spectral decrease, etc.)
4. Perceptual features use auditory-filtered bandwidth versions of the signal and aim at approximating the human perception of sounds using Equivalent Rectangular Bandwidth (ERB) scale [14] and gammatone filterbank [15] (e.g. loudness, ERB spectral centroid, etc.).

We implemented the proposed set of features in python according to the existing matlab timbre toolbox described in detail in [12].

Acronym	Feature name	#
Att	Attack duration (see ADSR model [16])	1
AttSlp	Attack slope (ADSR)	1
Dec	Decay duration (ADSR)	1
DecSlp	Decay slope (ADSR)	1
Rel	Release duration (ADSR)	1
LAT	Log Attack Time	1
Tcent	Temporal centroid	1
Edur	Effective duration	1
FreqMod, AmpMod	Total energy modulation (frequency,amplitude)	2
RMSenv	RMS envelope	2
ACor	Signal Auto-Correlation function (12 first coef.)	24
ZCR	Zero-Crossing Rate	2
HCent	Harmonic spectral centroid	2
HSprd	Harmonic spectral spread	2
HSkew	Harmonic skewness	2
HKurt	Harmonic kurtosis	2
HSlp	Harmonic slope	2
HDec	Harmonic decrease	2
HRoff	Harmonic rolloff	2
HVar	Harmonic variation	2
HErg, HNErg, HFErg,	Harmonic energy, noise energy and frame energy	6
HNois	Noisiness	2
HF0	Fundamental frequency F_0	2
HinH	Inharmonicity	2
HTris	Harmonic tristimulus	6
HodevR	Harmonic odd to even partials ratio	2
Hdev	Harmonic deviation	2
SCent, ECent	Spectral centroid of the magnitude and energy spectrum	4
SSprd, ESprd	Spectral spread of the magnitude and energy spectrum	4
SSkew, ESkew	Spectral skewness of the magnitude and energy spectrum	4
SKurt, EKurt	Spectral kurtosis of the magnitude and energy spectrum	4
SSlp, ESlp	Spectral slope of the magnitude and energy spectrum	4
SDec, EDec	Spectral decrease of the magnitude and energy spectrum	4
SRoff, ERoff	Spectral rolloff of the magnitude and energy spectrum	4
SVar, EVar	Spectral variation of the magnitude and energy spectrum	4
SFErg, EFErg	Spectral frame energy of the magnitude and energy spectrum	4
Sflat, ESflat	Spectral flatness of the magnitude and energy spectrum	4
Scre, EScre	Spectral crest of the magnitude and energy spectrum	4
ErbCent, ErbGCent	ERB scale magnitude spectrogram / gammatone centroid	4
ErbSprd, ErbGSprd	ERB scale magnitude spectrogram / gammatone spread	4
ErbSkew, ErbGSkew	ERB scale magnitude spectrogram / gammatone skewness	4
ErbKurt, ErbGKurt	ERB scale magnitude spectrogram / gammatone kurtosis	4
ErbSlp, ErbGSlp	ERB scale magnitude spectrogram / gammatone slope	4
ErbDec, ErbGDec	ERB scale magnitude spectrogram / gammatone decrease	4
ErbRoff, ErbGRoff	ERB scale magnitude spectrogram / gammatone rolloff	4
ErbVar, ErbGVar	ERB scale magnitude spectrogram / gammatone variation	4
ErbFErg, ErbGFErg	ERB scale magnitude spectrogram / gammatone frame energy	4
ErbSflat, ErbGSflat	ERB scale magnitude spectrogram / gammatone flatness	4
ErbScre, ErbGScre	ERB scale magnitude spectrogram / gammatone crest	4
Total		164

Table 1. Acronym, name and number of the used timbre descriptors.

2.2 Classification

This paper investigates three distinct supervised classification methods which are respectively the support vector machines (SVM), the convolutional neural network (CNN) and the densely connected neural network (DNN).

2.2.1 Support Vector Machines

The model based on Support Vector Machines (SVM) is a commonly-used machine learning technique which offers an efficient solution for supervised classification problems [17]. The principle of SVM consists of a projection of the data into a feature space with a higher dimension before computing a linear hyperplane separator trained from annotated examples. Due to their robustness, SVM are often used as a baseline classification method in comparison to more efficient techniques such as decision trees or artificial neural networks. However, it requires a suitable choice of the kernel function. In this paper, we use a Gaussian radial basis function $f(x,\hat{x}) = \exp(-\gamma||x-\hat{x}||^2)$ where γ is defined as one divided by the number of input features. In our study, we used the python implementation provided by Scikit-learn[1].

2.2.2 2D convolutional neural network

Convolutional neural networks (CNN, or ConvNet) is a popular methods which showed its efficiency in a large number of audio and image classification problems. In our study, we use the deep 2D CNN detailed in Table 2 inspired from [8] where the size of the input layer depends on the choice of the computed audio features (i.e MFCC or STFT). The input is then processed by 4 convolutional blocks (two layers with 16 filters of size 3×3 and two layers of 16 filters of size 3×1) including a batch normalization, max pooling layer of size 2×2 and a dropout of 25%. The output of the convolutional blocks is followed by 3 fully connected layers (256 units, 32 units and 1 unit). All the fully connected layers use a leaky rectifier activation function except for the output layer which uses the softmax function. We implemented this model in python using Keras[2].

2.2.3 Dense neural network

The proposed Dense neural network (DNN) is described in table 3. Its input layer corresponds to the 164 timbre features coefficients which are computed from the input signal according to Table 1. Then, it is processed by 4 fully connected layers of 328 neurons and a output layer of 2 neurons. All the layers use leaky rectifier as activation function with an exception for the output layer which uses the softmax function.

[1] https://scikit-learn.org
[2] https://keras.io/

#	Output size	Layer type	Filter size
1	$20 \times 44 \times 1$ or $512 \times 44 \times 1$	Input	
2	$20 \times 44 \times 16$	Convolutional	3×3
3	$20 \times 44 \times 16$	BatchNormalization	-
4	$10 \times 22 \times 16$	Max pooling	2×2
5	$10 \times 22 \times 16$	Convolutional	3×3
6	$10 \times 22 \times 16$	BatchNormalization	-
7	$5 \times 11 \times 16$	Max pooling	2×2
8	$5 \times 11 \times 16$	Convolutional	3×1
9	$5 \times 11 \times 16$	BatchNormalization	-
10	$3 \times 6 \times 16$	Max pooling	2×2
11	$3 \times 6 \times 16$	Convolutional	3×1
12	$3 \times 6 \times 16$	BatchNormalization	-
13	$2 \times 3 \times 16$	Max pooling	2×2
14	256	Fully connected	
15	32	Fully connected	
16	2	Fully connected	
17	2	Soft max	
18	1	Output	

Table 2. Architecture of the proposed 2D CNN.

#	Output size	Layer type
1	164×1	Input
2	164	Fully connected
3	328	BatchNormalization
4	328	Fully connected
5	328	BatchNormalization
6	328	Fully connected
7	328	BatchNormalization
17	2	Softmax
18	1	Output

Table 3. Architecture of the proposed DNN for which the input layer corresponds to the 164 proposed timbre features.

3 Evaluation

3.1 Dataset

Our experiments use the publicly available dataset[3] which was investigated in [8] and was acquired from 6 distinct beehives in the context of two different projects: the Open Source Beehive (OSBH) project and the NU-Hive project [4]. The dataset contains about 96 hours of audio signals which were recorded inside the beehive using a MicroElectrical-Mechanical System (MEMS) microphone, we only used 30 files from 576 (Hive 1, Hive 3) of the NU Hive project (3H20), and 20 files of 5 min collected from OSBH project(1H40). The audio signals are labeled as "active" and "missing queen" which correspond to the distinct labels to predict. Moreover, each audio signal is segmented between the distinct labels "bee" and "no bee" to discriminate the sounds produced by the bees from ambient noises. A threshold of 0.5 disregards intervals shorter than half a second, thus defining the minimum duration of the no bee intervals.

3.2 setup

As a pre-processing step, the audio signals are segmented in one-second-long homogeneous time frames (with the same label) and are resampled at $F_s = 22.05$ kHz to obtain 17,295 distincts individual where 8,444 are labeled as "active" and 8,851 are labeled as "missing queen". Two distinct experimental protocols are used to comparatively assess

[3] https://zenodo.org/record/1321278

the different investigated methods.

The first experiment uses a random split of the whole dataset and merges the data from the 6 beehives. This configuration uses 70% of the dataset as a training set and 30% as the test set.
The second experiment uses a 4-fold-cross validation where each fold corresponds to a distinct beehive except for four hives which are merged into two distinct folds (CF001+CF003 and CJ001+GH001) because they only contain "active" (resp. "missing queen" individuals).

In each experiment, exactly the same split is used to assess each method to make the result comparable. For the training step, our computations based on the neural networks use the RMSprop optimizer with a batch size equal to 145 and a number of epochs of 50. Our hardware configuration is based on an Intel Intel(R) Xeon(R) W-2133 CPU @ 3.60GHz CPU with 32GB or RAM and a NVIDIA GTX 1080 TI GPU.

3.3 Results

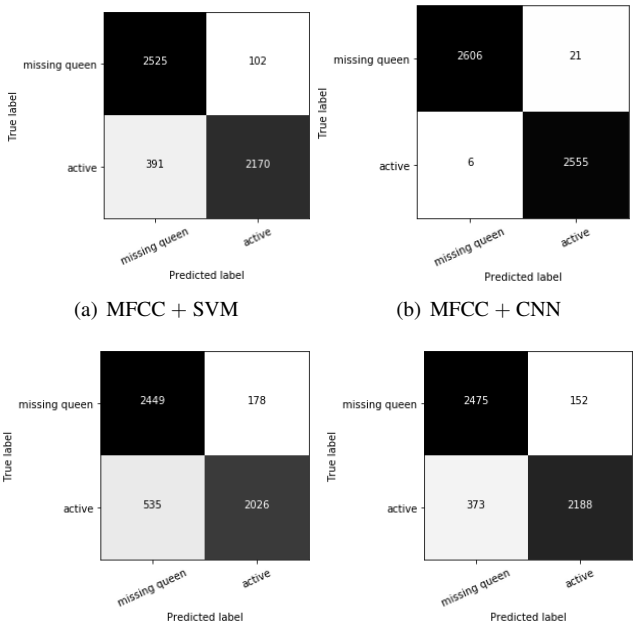
The results obtained for the two experiments expressed in terms of F-measure, Precision, Recall and Accuracy are presented in Tables 4 and 5. The detailed confusion matrices of the random split experiment is presented in Fig. 2. The results shows that the MFCC + CNN approach obtains the best results for the random split experiment which obtain an excellent average F-measure equal to 0.99 that is higher than 0.90 obtained with the timbre features combined with DNN and than the F-measures obtained with the STFT+CNN method which obtain the poorest results. Interestingly, the 4-fold-cross-validation experiment reveals a lack of generalization and a sensitivity problem of the MFCC-based trained models which obtain the poorest with an accuracy as low as 0.41 (MFCC+CNN). In this configuration, the STFT obtains the best results with an accuracy of 0.61. In both experiment, the timbre features obtains the most balanced results with a significantl lower number of parameters (164 real scalars) in comparison to MFCC (880) and STFT (22,528). This clearly shows an advantage of the timbre features to obtain good prediction using a very low number of computed features.

Method	Size	Label	F-measure	Prec.	Rec.	Acc.
MFCC+SVM	$\mathbb{R}^{20\times44}$	missing-queen	0.91	0.87	0.96	0.90
		active	0.90	0.85	0.90	
MFCC+CNN	$\mathbb{R}^{20\times44}$	missing-queen	0.99	1	0.99	0.99
		active	0.99	0.99	1	
STFT+CNN	$\mathbb{R}^{512\times43}$	missing-queen	0.30	0.97	0.18	0.58
		active	0.70	0.54	0.99	
Timbre feat.+SVM	$\mathbb{R}^{164}$	missing-queen	0.87	0.82	0.93	0.86
		active	0.85	0.92	0.79	
Timbre feat.+DNN	$\mathbb{R}^{164}$	missing-queen	0.90	0.87	0.94	0.90
		active	0.89	0.94	0.85	

Table 4. Comparative results obtained for the random split experiment.

Method	Size	Label	F-measure	Prec.	Rec.	Acc.
MFCC+SVM	$\mathbb{R}^{20\times44}$	missing-queen	0.12	0.09	0.45	0.09
		active	0.01	0.01	0.01	
MFCC+CNN	$\mathbb{R}^{20\times44}$	missing-queen	0.34	0.27	0.51	0.41
		active	0.38	0.417	0.57	
STFT+CNN	$\mathbb{R}^{512\times43}$	missing-queen	0.003	0.33	0.021	0.61
		active	0.76	0.87	0.86	
Timbre feat.+SVM	$\mathbb{R}^{164}$	missing-queen	0.25	0.25	0.39	0.31
		active	0.33	0.38	0.33	
Timbre feat.+DNN	$\mathbb{R}^{164}$	missing-queen	0.30	0.31	0.38	0.45
		active	0.54	0.6	0.7	

Table 5. Comparative results obtained for the 4-fold cross-validation experiment.



(a) MFCC + SVM (b) MFCC + CNN

(c) Timbre feat + SVM (d) Timbre feat. + DNN

Fig. 2. Confusion matrices obtained respectively with the MFCC+SVM (a), MFCC+CNN (b), Timbre feat.+SVM (c) and Timbre feat.+DNN (d) for the random split experiment.

4 Conclusion

We compared together several methods and we introduced a new one based on audio timbre features taken from the music information retrieval (MIR) literature for predicting the health state of a beehive. Our method shows an improvement of the model robustness when compared to several state-of-the-art methods. It allows to significantly reduce the number of required input size to 164 real scalars and paves the way of efficient embedded-system-based implementations. However, the cross-hive analysis results appear to be insufficient and will require a further investigation involving a larger annotated dataset. Future work will consist in a real-world integration of our approach using an embedded system and a consideration of more beehive state labels such as bees swarming or queen piping and quacking which are full of interest for beekeepers.

References

[1] Bryden, J., Gill, R. J., Mitton, R. A., Raine, N. E., and Jansen, V. A., 2013. "Chronic sublethal stress causes bee colony failure". *Ecology letters,* **16**(12), pp. 1463–1469.

[2] Booton, R. D., Iwasa, Y., Marshall, J. A., and Childs, D. Z., 2017. "Stress-mediated allee effects can cause the sudden collapse of honey bee colonies". *Journal of theoretical biology,* **420**, pp. 213–219.

[3] Cecchi, S., Terenzi, A., Orcioni, S., and Piazza, F., 2019. "Analysis of the sound emitted by honey bees in a beehive". In Audio Engineering Society Convention 147.

[4] Cecchi, S., Terenzi, A., Orcioni, S., Riolo, P., Ruschioni, S., and Isidoro, N., 2018. "A preliminary study of sounds emitted by honey bees in a beehive". In Audio Engineering Society Convention 144.

[5] Kulyukin, V. A., Mukherjee, S., Burkatovskaya, Y. B., et al., 2018. "Classification of audio samples by convolutional networks in audiobeehive monitoring". *Tomsk State University Journal of Control and Computer Science*(45), pp. 68–75.

[6] Kulyukin, V., Mukherjee, S., and Amlathe, P., 2018. "Toward Audio Beehive Monitoring: Deep Learning vs. Standard Machine Learning in Classifying Beehive Audio Samples". *Applied Sciences,* **8**(9), Sept., p. 1573.

[7] Cejrowski, T., Szymański, J., Mora, H., and Gil, D., 2018. "Detection of the Bee Queen Presence Using Sound Analysis". In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Vol. 10752 LNAI, pp. 297–306.

[8] Nolasco, I., Terenzi, A., Cecchi, S., Orcioni, S., Bear, H. L., and Benetos, E., 2019. "Audio-based identification of beehive states". In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp. 8256–8260.

[9] Tiwari, V., 2010. "Mfcc and its applications in speaker recognition". *International journal on emerging technologies,* **1**(1), pp. 19–22.

[10] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O., 2015. "librosa: Audio and music signal analysis in python". In Proceedings of the 14th python in science conference, Vol. 8, pp. 18–25.

[11] Robles-Guerrero, A., Saucedo-Anaya, T., González-Ramírez, E., and De la Rosa-Vargas, J. I., 2019. "Analysis of a multiclass classification problem by lasso logistic regression and singular value decomposition to identify sound patterns in queenless bee colonies". *Computers and Electronics in Agriculture,* **159**, pp. 69–74.

[12] Peeters, G., Giordano, B., Susini, P., Misdariis, N., and McAdams, S., 2011. "The timbre toolbox: Audio descriptors of musical signals". *Journal of Acoustic Society of America (JASA),* **5**(130), Nov., pp. 2902–2916.

[13] Schubert, E., Wolfe, J., and Tarnopolsky, A., 2004. "Spectral centroid and timbre in complex, multiple instrumental textures". In Proc. 8th Int. Conf. on Music Perception & Cognition (ICMPC).

[14] Moore, B., and Glasberg, B., 1983. "Suggested formulae for calculating auditory-filter bandwidths and excitation patterns". *Journal of the Acoustical Society of America,* **74**, pp. 750–753.

[15] E.Ambikairajah, Epps, J., and Lin, L., 2001. "Wideband speech and audio coding using gammatone filter banks". In Proc. IEEE ICASSP'01, pp. 773–776.

[16] Torelli, G., and Caironi, G., 1983. "New polyphonic sound generator chip with integrated microprocessor-programmable adsr envelope shaper". *IEEE Trans. on Consumer Electronics,* **CE-29**(3), pp. 203–212.

[17] Steinwart, I., and Christmann, A., 2008. *Support vector machines*. Springer Science & Business Media.