POLITECNICO DI TORINO
Repository ISTITUZIONALE

Audio-Based Identification of Queen Bee Presence Inside Beehives

Original
Audio-Based Identification of Queen Bee Presence Inside Beehives / Barbisan, Luca; Turvani, Giovanna; Riente, Fabrizio. - ELETTRONICO. - (2023), pp. 70-74. (Intervento presentato al convegno IEEE Conference on Agrifood Electronics tenutosi a Torino (Italy) nel 25-27 September 2023) [10.1109/CAFE58535.2023.10291679].

Availability:
This version is available at: 11583/2984526 since: 2023-12-14T20:40:29Z

Publisher:
IEEE

Published
DOI:10.1109/CAFE58535.2023.10291679

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
IEEE postprint/Author's Accepted Manuscript

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

11 February 2024

Audio-based Identification of Queen Bee Presence Inside Beehives

Luca Barbisan*, Giovanna Turvani* and Riente Fabrizio*
*Department of Electronics and Telecommunications, Politecnico di Torino, Torino, Italy

Email: fabrizio.riente@polito.it

***Abstract*—Honeybees are essential for the health of people and the planet. They play a key role in the pollination of most crops. The high mortality observed in the last decade, caused by stress factors among which the climate change, have raised the necessity of remote sensing the beehives to help monitor the health of honeybees and better understand this phenomenon. Several solutions have been proposed in the literature, and some of them include the analysis of in-hive sounds. In this scenario, we explore the potential of machine learning methods for queen bee detection using only the audio signal, being a good indicator of the colony state of health. In particular, we experiment support vector machines and neural network classifiers. We consider the effect of varying the audio chunk duration and the adoption of different hyperparameters.**

***Index Terms*—Beekeeping, agriculture, internet of things, environmental monitoring, sensor networks, sound analysis**

I. Introduction

Honeybees are of vital importance to global crop production. They pollinate 70 of the around 100 crop species that feed 90% of the world. Colony losses have reached historic levels all over the world during the last decades. The causes of these phenomena can be found in the so-called "colony collapse disorder" (CCD) [1]. Climate change, intensive agriculture, land-use change, pesticides, biodiversity loss, Varroa mites, and pollution are the leading cause of bees' death worldwide. The decline in honeybee health has resulted in beekeepers' and researchers' demand for novel mechanisms of monitoring colony health. In this scenario, intensive beehive monitoring is necessary to better understand this phenomenon and try to help these important insects. In the literature, several approaches have been proposed. Most of them are based on sensing several hive parameters such as temperature, humidity, carbon dioxide and weight [2], [3], [4]. Other approaches exploited computer vision to track bees at the hive entrance [5], [6]. Some others showed that embedded systems based on Long Range communication protocols look very promising [3], [7]. Despite all these approaches showing good results, some of them require a processing unit with remarkable computational power and in some cases require modification of the beehive [5], [6], [2], making the solution impractical for realistic applications. A real system must be compact, composed of environmental sensors and microphones, and it should be energy efficient and able to run AI models in short time. Recent studies showed that sound analysis appears to be the most promising technique for this purpose, particularly, as non-invasive solution. Indeed, bees make many different sounds, they use a combination of vibroacoustic signal to communicate [8]. Researchers showed that sound analysis can enable the detection of swarming [9], the presence of young queen or the presence of absence of the queen within the colony [10]. The queen presence is an important indicator of the colony health [8]. Timely notification of such events are of extremely importance for beekeepers. In this context, solutions exploiting convolutional neural networks (CNNs) and support vector machines (SVMs) have been presented to classify the queen presence from audio recordings. One example has been reported in [11]. In particular, they explored features based on Mel Frequency Cepstral Coefficients (MFCCs), short time Fourier Transform (STFTs) and Hilbert-Huang transform (HHT). In this paper, we started from the above considerations and we analyzed the performance of Neural Networks (NNs) and SVM classifiers varying the complexity of the model with the aim of detecting the queen presence. The paper is organized as follows: In section II, we introduce the proposed approach describing the used dataset, the developed machine learning framework from feature extraction to the classification. In section III, we present the results comparing the different models. Finally, section IV concludes the paper.

II. Proposed Approach

The proposed approach relies solely on analyzing the audio captured within the beehive. This non-invasive solution enables continuous monitoring of the beehive, without disturbing the bees or altering their natural behavior. It can provide valuable information for beekeepers, researchers, or conservationists, helping them make informed decisions about hive management, health assessments, or interventions if necessary.

The sound level amplitude and frequency depend on the colony activity and health state. Flying bees produce a typical sound with an approximate frequency of 250Hz, worker bees produce a characteristic frequency that lies between 225Hz and 285Hz to cool the hive with ventilation, and the frequency can reach up to 3000Hz as a defensive reaction to potential outside threats to the colony [12] [13]. However, if the queen is missing in the beehive the bees produce a sound that contains harmonics with lower frequencies [14]. The proposed approach can be divided into five steps: i) analysis of the dataset; ii) audio chunk extraction; iii) feature extraction; iv) compute model statistics; v) generate the final model and

vi) test the model evaluating its performance. A detailed description of each step is reported in the following sections.

A. Dataset

In this work, the ”Smart Bee Colony Monitor: Clips of Beehive Sounds” [15] is used for training and evaluating the proposed system. It is a dataset with 7100 audio and environment data acquisitions of European honey bee hives in California. Each file is a 60 seconds audio sound, recorded at 22050Hz with a depth of 32 bits floating point for a single channel. In addition to audio data, each sample is associated with internal condition of the beehive and weather conditions [15]. For the purpose of this study, we only used the recorded audio (removing saturation artifacts) and the queen presence label.

B. Audio chunk split

In this study, we implemented a data augmentation technique to increase the number of input samples by splitting audio files into smaller chunks. With this approach, it is possible to increase the number of chunks changing the hop size that is the space between the start of subsequent chunks inside the original audio. We then compare the impact on the results by examining the use of different chunk sizes: 0.5, 1, 3, 5 seconds but with the same number of chunks and no overlapping. For this purpose we used the hop size equal to 5 and we obtain 78100 samples for each experiment.

C. Feature extraction

Raw audio files contains a large number of time samples and would be difficult to use them directly as input for the classification. Therefore, it is necessary to extract a compact and meaningful representation using a reduced number of coefficients. For this purpose, we analyzed two of the most common feature extraction used for audio applications and in particular bring the best results in the field of bees monitoring.

The first is the MFCC, which is a commonly used audio feature extraction technique in the field of audio processing. It aims to capture the relevant characteristics of the human auditory system by converting the audio signal into a compact representation with a n selected number of coefficients. In this work, we compared the results with n ranging from 10 to 50. The MFCC extraction is composed of the following steps applied to each dataset sample:

1) The audio is divided in windows of 2048 audio samples partially overlapped using a hop length of 512 samples.
2) The Hann windowing function is used to smoothing the signal at the edges of the window.
3) Discrete Fourier Transform (DFT) is applied to each window to convert the signal to the frequency domain.
4) Apply the Mel Filterbank set of triangular filters evenly spaced on the Mel scale.
5) Discrete Cosine transform (DCT) is applied in order to obtain the coefficients for each window.
6) The final step computes for each coefficient the mean value among all the windows. With this operation we obtain the n number of features (indicated with MFCC n) that are used as input for the classifiers.

The second approach uses the Short Time Fourier Transform (STFT) that simplify the required steps in order to reduce the computation complexity that is required:

1) The audio is divided in windows ranging from 256 to 4096 audio samples without overlapping.
2) The Hann windowing function is used to smoothing the signal at the edges of the window.
3) DFT is applied to each window to convert the signal to the frequency domain.
4) The magnitude of the spectrogram is computed.
5) The final step computes, for each frequency bin, the mean value among all the window. With this final step we obtain a number of inputs for the classifier, which are function of the window size ($f_{bins} = \frac{m}{2} + 1$), where m is the window size (indicated with STFT m).

D. Classifiers

We compared two different classifiers for the task of binary classification: a neural network implemented using the Keras library [16] and a support vector machine (SVM) implemented using the scikit-learn (sklearn) library [17].

1) NN classifier: The neural network is a powerful and flexible machine learning model that can learn complex patterns in the data. We constructed a multi-layer perceptron architecture, consisting of multiple fully connected layers, tuned the layers and the number of neurons. We used the ReLu activation function for the inner layers and a sigmoid function for the output layers then converted to a binary value using a threshold of 0.5. We trained the network using early stopping with patience of 5 to prevent overfitting and to determine the optimal number of epochs.

2) SVM classifier: SVM is a widely used and effective supervised learning algorithm for classification tasks. It works by finding an optimal hyperplane that maximally separates the data points belonging to different classes. For the implementation of SVM, we utilized the scikit-learn (sklearn) library. We experimented with RBF (Radial Basis Function) kernel function and tuned the C parameter.

E. Experimental setup

The proposed approach explored different chunk sizes with different classifiers. The schematic representation of the experimental setup adopted for every configuration is reported in Fig. 1. After dividing the dataset into chunks for data augmentation, we proceeded with the dataset split into a training set and a test set. The training set comprised 80% of the data, while the remaining 20% was allocated to the test set. This division ensured a sufficient amount of data for training while preserving a separate portion for the final test of the selected model.

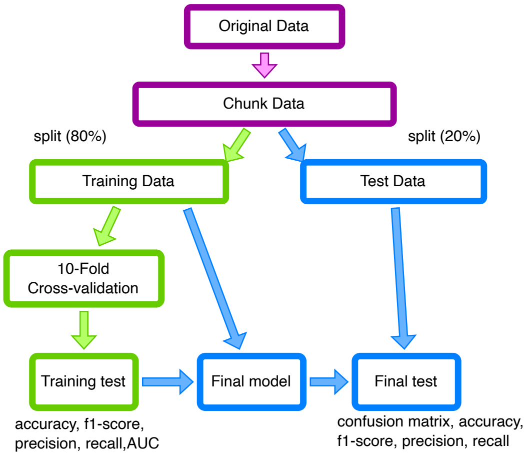

Fig. 1: Diagram of experimental setup adopted in the proposed approach for each configuration of the classifier and chunk configuration.

To assess the generalization performance of the models and obtain reliable statistics, we applied the 10-fold cross-validation technique on the training set. This technique involves dividing the training data into 10 equally sized folds. The models were trained and evaluated 10 times, with each fold serving as the validation set once while the remaining nine folds are used for the training. This process allowed us to estimate how well the models can generalize to unseen data and mitigate biases introduced by a particular training-test split. At the end of each experiment, we proceeded to train the final model using the entire training dataset. This final model was trained on the combined data from all 10 folds, providing a larger sample for training and potentially improving the model's performance.

The performance evaluation of the final model was conducted using the test dataset that was previously set aside. This separate test dataset allowed us to assess the actual behavior of the exported model on unseen data, providing an unbiased estimate of its performance in real-world scenarios.

By employing the 10-fold cross-validation technique during training and conducting a final test on a separate test dataset, we aimed to obtain reliable performance metrics for evaluating the models' effectiveness in both generalizations and real-world scenarios.

F. Evaluation metrics

In the experimental phase, we extracted different metrics for both the training and the final test evaluations. These metrics were selected to capture various aspects of the model's performance and cover a wide range of evaluation criteria. For the Training Test, we extracted the mean value and standard deviation of the results obtained by the 10 iterations of the cross-validation: Accuracy, Precision, Recall, f1-score, Area Under the Curve (AUC). For the Final Test, the Accuracy, Precision, Recall, f1-score, and the Confusion Matrix were considered. However, for the sake of clarity only the f1-score and Accuracy are reported in the following.

III. Results

The following experiments are performed considering 22050Hz as audio sample rate, window size 5s and hop size 5s. We varied the MFCC coefficients from 10 to 50 and the STFT window size from 256 to 4096. Table I and Table II report the scores of the training and test for both classifiers considering chunk sizes from 0.5 to 5 seconds, hop size = 5s to have 78100 samples. The configuration of the models are for NN (hidden layers=16 ReLU, 8 ReLU, epoch=500, batch size=64, learning rate=0.001, early stopping), for SVM (C=1, kernel=RBF) and the input features selected are MFCC 20 and STFT 2048. The metrics show that larger chunk sizes (3s-5s) perform better with both classifiers. We selected a chunk size of 5s for the subsequent analysis.

TABLE I: SVM accuracy and f1-score with different chunk sizes.

chunk size	**feature**	**accuracy**	**accuracy test**	**f1-score**	**f1-score test**
0.5	mfcc20	0.9102 ± 2.93e-03	0.9124	0.9459 ± 1.95e-03	0.9472
1	mfcc20	0.9321 ± 2.41e-03	0.9360	0.9595 ± 1.49e-03	0.9618
3	mfcc20	0.9536 ± 3.18e-03	0.9585	0.9726 ± 1.93e-03	0.9755
5	mfcc20	0.9593 ± 1.89e-03	**0.9591**	0.9761 ± 1.17e-03	**0.9759**
0.5	stft2048	0.8760 ± 8.90e-03	0.8833	0.9239 ± 5.79e-03	0.9285
1	stft2048	0.9024 ± 8.20e-03	0.9073	0.9408 ± 5.21e-03	0.9438
3	stft2048	0.9209 ± 9.42e-03	**0.9256**	0.9524 ± 5.88e-03	**0.9552**
5	stft2048	0.9210 ± 6.83e-03	0.9218	0.9524 ± 4.16e-03	0.9529

TABLE II: NN accuracy and f1-score with different chunk sizes.

chunk size	**feature**	**accuracy**	**accuracy test**	**f1-score**	**f1-score test**	
0.5	nn	mfcc20	0.9437 ± 3.13e-03	0.9467	0.9678 ± 1.92e-03	0.9694
1	nn	mfcc20	0.9566 ± 3.32e-04	0.9637	0.9751 ± 1.91e-04	0.9791
3	nn	mfcc20	0.9728 ± 1.10e-03	0.9767	0.9844 ± 6.56e-04	0.9865
5	nn	mfcc20	0.9743 ± 5.66e-04	**0.9821**	0.9852 ± 3.33e-04	**0.9897**
0.5	nn	stft2048	0.9500 ± 2.80e-03	0.9575	0.9711 ± 1.63e-03	0.9754
1	nn	stft2048	0.9713 ± 1.45e-04	0.9754	0.9835 ± 1.29e-05	0.9858
3	nn	stft2048	0.9824 ± 8.71e-04	0.9862	0.9899 ± 4.69e-04	0.9920
5	nn	stft2048	0.9904 ± 7.02e-04	**0.9930**	0.9945 ± 3.88e-04	**0.9960**

The results of using different input features (MFCC or STFT) are reported in Fig. 2(c-d) and summarized, numerically in Table III for the NN and Table IV for SVM classifiers. Increasing the number of MFCC coefficients and the window size of STFT give better results. However, this would increase the amount of the input features to be handled and therefore the complexity of the model. Our results show a higher f1-score (> 0.98) when compared with the CNN model with MFCC 20 adopted in [18] (f1-score = 0.9240), with both NN and SVM classifiers. In general, by looking at Table III-IV the NN models performed better than SVM, with the same configuration adopted for Table I-II. In particular, high f1-scores are achieved with low STFT window size (256 and 512). This is interesting, because a window size of 2048 is usually used, a it reduces the number of features at the input of the classifier. In the following, we analyzed the effect of different values of the regularization factor C for both MFCC 20 and STFT 2048. The windows size = 2048 is selected to be coherent the analysis in [18]. In both cases, we obtained high scores with larger C. Finally, we explored the effect of different configurations of the NN, with the same input features. We varied the layers from 1 to 4 and the number of neurons for each layer from 2 to 8 for the hidden layers.

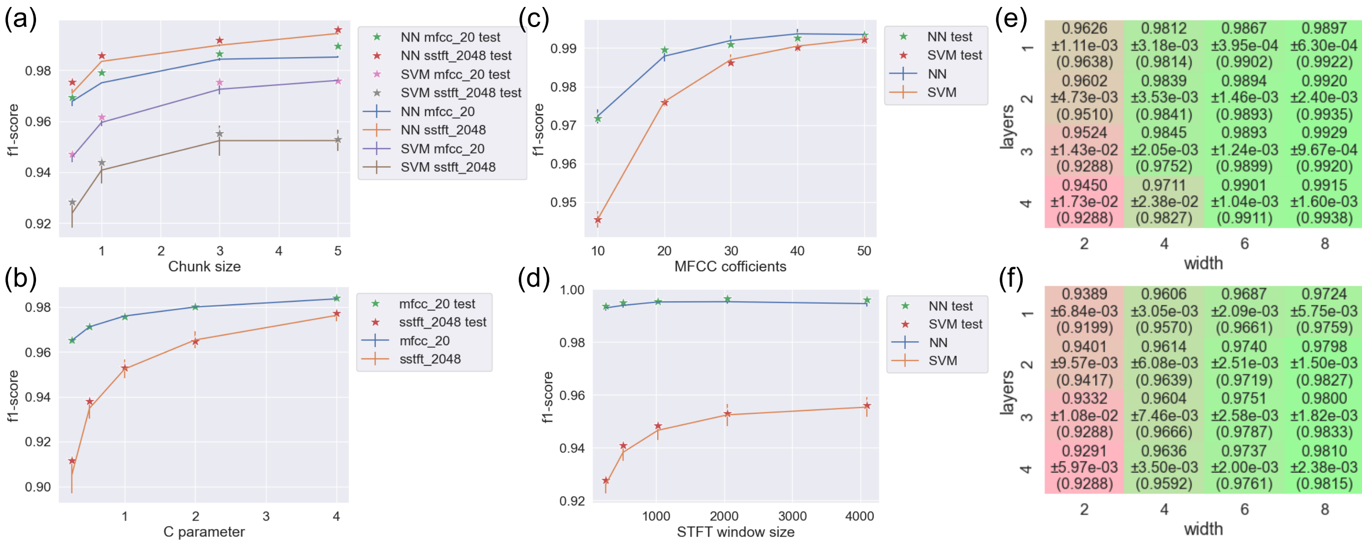

Fig. 2: Summary of the results: (a) with different chunk sizes; (b) with different regularization factor for the SVM classifier; (c) with different MFCC coefficients; (d) with different STFT window size; (e) f1-score, standard deviation and test results between parenthesis with respect to the number of neurons present in each layer of the neural network considering STFT 2048 as features; (f) same as (e) considering MFCC 20 features.

TABLE III: Accuracy and f1-score with NN and different features.

feature parameter	accuracy	accuracy test	f1-score	f1-score test
mfcc 10	0.9518 ± 3.01e-03	0.9509	0.9723 ± 1.84e-03	0.9718
mfcc 20	0.9790 ± 2.32e-03	0.9819	0.9879 ± 1.33e-03	0.9896
mfcc 30	0.9860 ± 2.18e-03	0.9843	0.9919 ± 1.26e-03	0.9909
mfcc 40	0.9891 ± 2.09e-03	0.9874	0.9937 ± 1.21e-03	0.9927
mfcc 50	0.9888 ± 1.27e-03	**0.9886**	0.9935 ± 7.14e-04	**0.9934**
stft 256	0.9878 ± 1.70e-03	0.9889	0.9930 ± 9.83e-04	0.9936
stft 512	0.9895 ± 1.63e-03	0.9912	0.9939 ± 9.47e-04	0.9949
stft 1024	0.9917 ± 9.74e-04	0.9920	0.9952 ± 5.70e-04	0.9954
stft 2048	0.9919 ± 1.23e-03	**0.9939**	0.9953 ± 7.15e-04	**0.9965**
stft 4096	0.9906 ± 2.00e-03	0.9934	0.9946 ± 1.16e-03	0.9962

TABLE IV: Accuracy and f1-score with SVM and different features.

feature param	accuracy	accuracy test	f1-score	f1-score test
mfcc 10	0.9097 ± 3.13e-03	0.9098	0.9455 ± 2.18e-03	0.9456
mfcc 20	0.9593 ± 1.89e-03	0.9591	0.9761 ± 1.17e-03	0.9759
mfcc 30	0.9778 ± 2.39e-03	0.9763	0.9870 ± 1.43e-03	0.9862
mfcc 40	0.9837 ± 2.40e-03	0.9833	0.9905 ± 1.40e-03	0.9903
mfcc 50	0.9869 ± 2.16e-03	**0.9866**	0.9924 ± 1.27e-03	**0.9922**
stft 256	0.8799 ± 5.42e-03	0.8825	0.9260 ± 3.45e-03	0.9277
stft 512	0.8986 ± 5.30e-03	0.9028	0.9382 ± 3.27e-03	0.9409
stft 1024	0.9117 ± 5.87e-03	0.9147	0.9465 ± 3.58e-03	0.9484
stft 2048	0.9210 ± 6.83e-03	0.9218	0.9524 ± 4.16e-03	0.9529
stft 4096	0.9256 ± 6.18e-03	**0.9268**	0.9553 ± 3.75e-03	**0.9561**

The f1-scores with their standard deviation are summarized in Fig. 2(e) for STFT 2048 and Fig. 2(f) for MFCC 20 features, respectively. The NN performed better, with similar scores with 6-8 neurons per hidden layer. On the contrary, the number of layers seems not to impact heavily on the network performance.

IV. Conclusion

This study investigated the detection of queen bee presence in the colony using SVM and NN varying both input features, chunk size and classifier parameters. Results indicated that

TABLE V: Accuracy and f1-score of SVM with different regularization factors.

C	feature	accuracy	accuracy test	f1-score	f1-score test
0.25	mfcc	0.8489 ± 1.25e-02	0.8587	0.9051 ± 8.22e-03	0.9117
0.5	mfcc	0.8935 ± 7.31e-03	0.8986	0.9348 ± 4.55e-03	0.9381
1	mfcc	0.9210 ± 6.83e-03	0.9218	0.9524 ± 4.16e-03	0.9529
2	mfcc	0.9418 ± 6.32e-03	0.9410	0.9654 ± 3.77e-03	0.9649
4	mfcc	0.9598 ± 4.49e-03	**0.9614**	0.9763 ± 2.63e-03	**0.9773**
0.25	stft	0.9413 ± 1.95e-03	0.9417	0.9651 ± 1.25e-03	0.9654
0.5	stft	0.9513 ± 1.62e-03	0.9515	0.9712 ± 1.04e-03	0.9713
1	stft	0.9593 ± 1.89e-03	0.9591	0.9761 ± 1.17e-03	0.9759
2	stft	0.9660 ± 1.51e-03	0.9665	0.9801 ± 9.46e-04	0.9803
4	stft	0.9721 ± 1.89e-03	**0.9727**	0.9837 ± 1.14e-03	**0.9841**

both SVM and NN models exhibited promising performance in accurately identifying the presence of a queen. However, the performance of these classifiers relies on the selection of appropriate parameters. Their tuning significantly improved the classification accuracy of both models. Therefore, the selection and optimization of classifier parameters are crucial for achieving optimal results. Overall, this study demonstrated that the detection of the queen bee can be successfully achieved using SVM and NN models. Audio chunk ranging between 3 to 5 seconds appears to be promising. The MFCC 20 features, as reported in [18] is a good trade-off between model complexity and accuracy. Similarly, the STFT with low dimensionality (256-512) showed good performance with the NN model. However, it is clear the parameters' selection significantly influenced the classification performance and larger parameters (chunk sizes, MFCC and STFT coefficients) could result better models. However, the parameters selection has been done with the aim of keeping the model simple for future deployment on a microcontroller.

REFERENCES

[1] J.-P. Faucon, L. Mathieu, M. Ribiere, A.-C. Martel, P. Drajnudel, S. Zeggane, C. Aurieres, and M. F. A. Aubert, "Honey bee winter mortality in france in 1999 and 2000," *Bee World*, vol. 83, no. 1, pp. 14–23, 2002.

[2] C. Stefania, S. Spinsante, T. Alessandro, and O. Simone, "A smart sensor-based measurement system for advanced bee hive monitoring," *Sensors*, vol. 20, no. 9, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/9/2726

[3] F. Bellino, G. Turvani, U. Garlando, and F. Riente, "An integrated multi-sensor system for remote bee health monitoring," in *2022 IEEE Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, 2022, pp. 334–338.

[4] A. Rafael Braga, D. G. Gomes, R. Rogers, E. E. Hassler, B. M. Freitas, and J. A. Cazier, "A method for mining combined data from in-hive sensors, weather and apiary inspections to forecast the health status of honey bee colonies," *Computers and Electronics in Agriculture*, vol. 169, p. 105161, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169919317661

[5] T. N. Ngo, K.-C. Wu, E.-C. Yang, and T.-T. Lin, "A real-time imaging system for multiple honey bee tracking and activity monitoring," *Computers and Electronics in Agriculture*, vol. 163, p. 104841, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0168169919301498

[6] C. Yang and J. Collins, "A model for honey bee tracking on 2d video," in *2015 International Conference on Image and Vision Computing New Zealand (IVCNZ)*, 2015, pp. 1–6.

[7] S. Kontogiannis, "An internet of things-based low-power integrated beekeeping safety and conditions monitoring system," *Inventions*, vol. 4, no. 3, 2019. [Online]. Available: https://www.mdpi.com/2411-5134/4/3/52

[8] Kirchner, W. H., "Acoustical communication in honeybees," *Apidologie*, vol. 24, no. 3, pp. 297–307, 1993. [Online]. Available: https://doi.org/10.1051/apido:19930309

[9] D. G. Dietlein, "A method for remote monitoring of activity of honeybee colonies by sound analysis," *Journal of Apicultural Research*, vol. 24, no. 3, pp. 176–183, 1985.

[10] H. Eren, L. Whiffler, and R. Manning, "Electronic sensing and identification of queen bees in honeybee colonies," in *IEEE Instrumentation and Measurement Technology Conference Sensing, Processing, Networking. IMTC Proceedings*, vol. 2, 1997, pp. 1052–1055 vol.2.

[11] A. Terenzi, N. Ortolani, I. Nolasco, E. Benetos, and S. Cecchi, "Comparison of feature extraction methods for sound-based classification of honey bee activity," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 112–122, 2022.

[12] A. Žgank, "Acoustic monitoring and classification of bee swarm activity using mfcc feature extraction and hmm acoustic modeling," in *2018 ELEKTRO*, 2018, pp. 1–4.

[13] W. Kirchner, "Acoustical communication in honeybees," *Apidologie*, vol. 24, no. 3, pp. 297–307, 1993.

[14] A. Terenzi, S. Cecchi, S. Orcioni, and F. Piazza, "Features extraction applied to the analysis of the sounds emitted by honey bees in a beehive," in *2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 2019, pp. 03–08.

[15] A. Yang, "Smart bee colony monitor: Clips of beehive sounds," 2022. [Online]. Available: https://www.kaggle.com/dsv/4451415

[16] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[17] 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*, vol. 12, pp. 2825–2830, 2011.

[18] A. Terenzi, S. Cecchi, and S. Spinsante, "On the importance of the sound emitted by honey bee hives," *Veterinary Sciences*, vol. 7, no. 4, 2020. [Online]. Available: https://www.mdpi.com/2306-7381/7/4/168