IoT-Based Bee Colony Health Monitoring: A Focus on Energy Impact and Audio Feature Extraction

(Article begins on next page)

IoT-Based Bee Colony Health Monitoring: A Focus on Energy Impact and Audio Feature Extraction

1st Andrea De Simone
Dept. of Elec. and Telecomm.
Politecnico di Torino
Torino, Corso Duca degli Abruzzi 24
andrea.desimone@polito.it

2nd Luca Barbisan
Dept. of Applied Science and Technology
Politecnico di Torino
Torino, Corso Duca degli Abruzzi 24
luca.barbisan@polito.it

3rd Giovanna Turvani
Dept. of Elec. and Telecomm.
Politecnico di Torino
Torino, Corso Duca degli Abruzzi 24
giovanna.turvani@polito.it

4th Fabrizio Riente
Dept. of Elec. and Telecomm.
Politecnico di Torino
Torino, Corso Duca degli Abruzzi 24
fabrizio.riente@polito.it

***Abstract*—Beekeeping is essential for maintaining ecosystem stability, improving pollination, and enriching biodiversity. The presence of the queen bee is critical for evaluating the health of bee colonies, which is fundamental for ecological equilibrium. Tine Machine Learning (TinyML), designed to predict the queen bee's presence, can be a valuable support for beekeepers in proactively assessing the colony's health from sound processing. Nonetheless, preprocessing the raw audio signals to prepare features for the TinyML model demands significant computational resources. This paper analyzes the energy impact of audio feature extractions for Tiny Machine Learning (TinyML) applications tailored for edge devices. We consider the Short Time Fourier Transform (STFT) and Mel-Frequency Cepstral Coefficients (MFCC) as audio features. Here, we deploy several models on our custom IoT device to detect the queen bee's presence from audio recordings. The system's architecture employs simplified Machine Learning models and low-precision audio processing to optimize energy efficiency. Despite these modifications aimed at reducing power usage, the system maintains performance metrics close to more complex setups, demonstrating minimal compromise in accuracy. This study underscores the impact of efficient feature extraction and data precision on reducing energy demands in IoT devices, which is crucial for sustainable TinyML deployment.**

***Index Terms*—Tiny machine learning (TinyML), internet of things (IoT), sound analysis, artificial intelligence (AI), neural network (NN), Queen bee detection**

I. Introduction

Honeybees are vital for pollination and the reproductive cycles of ecosystems, making them essential to the environment [1]. However, in recent years, bee populations have been declining, underscoring the fragility and importance of this valuable species [2]. This decline is attributed to factors such as habitat loss, pesticide exposure, climate change, and diseases [3] [4]. From 2003, researchers from many countries around the world are trying to understand the reasons behind the phenomenon known as Colony Collapse Disorder (CCD), which is strongly affecting the bee colonies in Europe and the rest of the world [5] [6]. Recent reports estimated that winter losses range between 22% and 36% in the United States [7]. Their decline poses a serious threat to agricultural productivity and ecological balance. Having a deeper understanding of the stress factors, the research community can devise targeted interventions to mitigate their impact.

In this context, beekeepers play a key role in mitigating the impact of diseases and supporting bees' dietary needs by planting flora that offers pollen, propolis, and nectar, and by ensuring the availability of water sources for colony development [8]. In addition, bee farms are often situated in remote, difficult-to-access locations, requiring beekeepers to travel extensively to visit and monitor their hives, a challenge particularly evident in nomadic beekeeping.

These considerations motivate the development of innovative solutions that can serve as support to researchers and beekeepers in their activities [3] [9] [10]. Indeed, several works have been presented in the literature to deepen the comprehension of the factors that contribute to the mortality of bees. This effort is crucial for protecting the health and sustainability of honey bee populations, which play a vital role in pollination and maintaining biodiversity. Some proposed systems perform real-time digital video processing [11], counting the number of bees entering and exiting the hive to understand the colony's activities. However, sound analysis alone has proven to be highly effective in detecting various hive events, such as swarming [12] and the presence or absence of a queen [13]. The presence of the queen bee is crucial for the colony's survival. Monitoring and predicting the queen's presence provides valuable insights into the overall health of the colony. Machine learning techniques can achieve high accuracy, close to 99%, in detecting the queen bee by analyzing the frequency spectrum of raw audio collected from within the hive. Commonly used coefficients for this purpose include Mel-frequency Cepstral Coefficients (MFCCs), Mel-spectrogram, and Short-Time Fourier Transform (STFT) [14]–[17]. Most proposed works focus solely on developing accurate machine learning models, without considering their complexity when deployed on an embedded device. In [15] do the authors concentrate on creating models with limited parameters, comparing the accuracy of STFT and MFCC coefficients across different networks and audio duration. Extracting STFT and MFCC features requires a high computational effort for a resource-limited embedded system. To our knowledge, prior work has yet to analyze the impact of feature extraction (STFT or MFCC) on the energy consumption of an embedded system integrated into the beehive. In this paper, we investigate the impact of this pre-processing steps, fundamental for the prediction of the queen bee presence. We present the analysis performed on a custom designed system, composed of a STM32WLxx MCU. We measured the power consumption for the feature extraction step, which is some order of magnitude larger than the inference. The paper is organized as follows: In section II, the metholodogy adopted for the study is presented. In section III, we recall the processing steps to extract the STFT and MFCC features. Section IV presents the measurements performed on our embedded system on different models using a variable number of STFT and MFCC coefficients. Finally, section V concludes the paper.

II. Methodology

Two open-source datasets have been selected from the works present in the literature. These include audio samples recorded inside the beehive, annotated with the queen bee presence state. Dataset I [18] includes 7,100 minutes of audio, and Dataset II [19] contains 5,730 minutes of recordings. MFCCs or STFT audio coefficients are extracted from the datasets and are employed in combination with a machine learning model to predict if the queen bee is present inside the beehive. In the early stage, a Python framework is developed to extract the audio coefficients in floating-point format. After a scaling and feature selection process, these coefficients are then utilized as features to train machine learning models. The type of model selected for this work is a neural network consisting of two fully connected layers and a final layer terminated by a sigmoid activation function to produce the binary decision. The models are built, trained and tested with the TensorFlow library. The number of outputs of the fully connected layers is chosen to be small, in order to generate networks with less than 3,000 weights. Despite the extremely tiny models, they achieve an accuracy of over 95%. This choice is made to minimize the computational effort and the memory requested to store the weights in the microcontroller. Compared to the results exposed in the literature, where networks generally are more complex and have millions of parameters, this small drop in performance is a good compromise to enable the execution of the inference on the edge. This initial configuration is set entirely in floating-point and cannot yet be executed in the microcontroller, an overview is displayed in Fig. 1 (a). This step was employed to configure the model parameters and the accuracies were taken as reference for the successive phase.

For the STFT algorithm, we tested how the number of coefficients used as features influences the accuracy of the model, indeed, high-frequency coefficients generally includes less useful information for the prediction. Different combinations of FFT size and number of coefficients retained are tested during this phase. For MFCCs extraction the parameters that can be modified are the FFT size and the number of mel-frequencies. MFCCs provide better audio information compression but require a higher computational effort for the extraction. Usually, 10-20 coefficients are enough to achieve good accuracy. On the other hand, STFT has the opposite properties, and generally, several hundred or thousands of coefficients are required for the machine learning model.

Another variable tested is the length of the audio frame. A shorter recording means fewer audio samples, resulting in lower energy requirements for extraction but also less information to be used for inference. From the analysis performed in [15], we observed that with 3 seconds, a good level of accuracy is reached. Increasing the length to 5 seconds leads to a small improvement in the score with limited additional energy required. Recording for more than 5 seconds does not significantly improve the scores for the network; for this reason, longer audio recordings are not considered in this study.

The feature extraction process is transposed into C code in fixed-point format, using *int16* or *int32*, as illustrated in Fig. 1 (b). The network is quantized with TensorFlow Lite utilizing post training quantization technique and then is converted in C code using X-CUBE-AI, an extension package provided by ST Microelectronics. This package provides a command line interface for converting a Python model and emulating the behavior of the resultant C network. This interface is integrated in the Python framework and is exploited to set up a new flow of operation where all the computation is performed in fixed-point. The resultant metrics of the new model is compared to the initial floating-point approach to assess the effect of the quantization and the integer feature extraction.

This configuration can be actually deployed in the microcontroller. A custom board is developed to be inserted inside a beehive for predicting the state of the queen bee's presence and send the information remotely. The board includes the microcontroller, a MEMS digital microphone and an additional flash memory to store the collected audios. The system is powered by a battery, so it is fundamental to limit the energy consumption to maximize the battery life.The microcontroller records a new audio with a certain duty cycle, during this interval it enters in a low-power mode that consumes only a few microamperes. Subsequently, the coefficients (STFT or MFCCs) are extracted and after a pre-processing (which includes scaling and feature selection), they can be used as features for the machine learning model. The state of the queen bee's presence is predicted and finally, the information is transmitted by an on-board antenna using the LoRaWAN protocol. Since the network is small, the inference operation is not critical in terms of energy consumption, indeed, it takes only few milliseconds. On the other hand, the extraction process requires several seconds to be completed. For this reason, the focus is placed on audio processing in the following part of this paper.

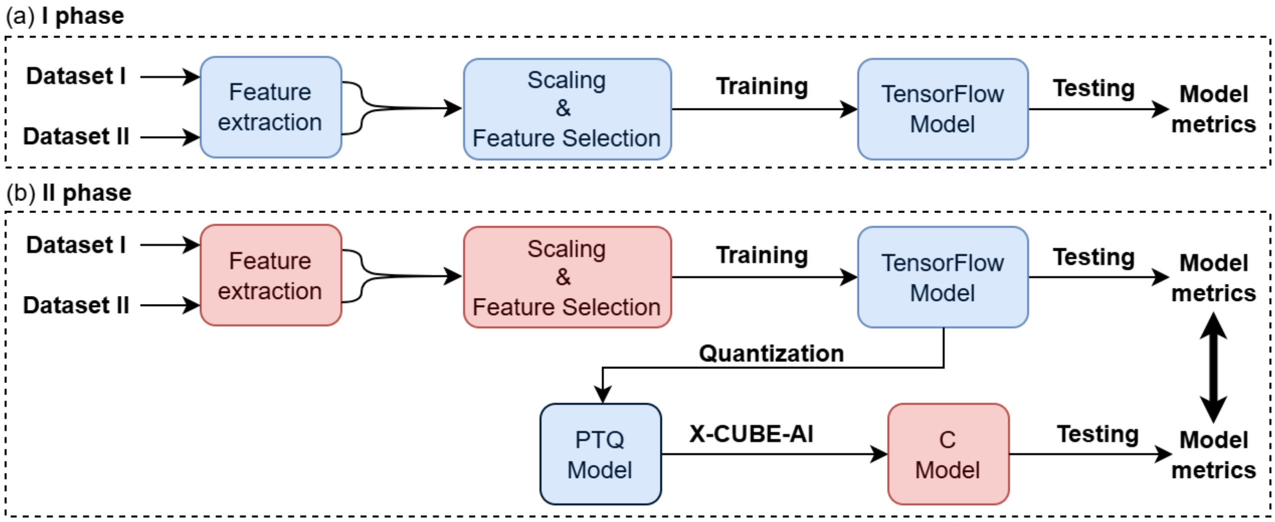

Fig. 1: Diagram of the methodology adopted: the blue rectangles represent the floating-point operations, the red rectangles represent the fixed-point operation that can be executed in the microcontroller. (a) displays the first phase completely in Python, (b) details the second phase where there is the conversion and transposition in C.

III. Feature Extraction and Models Selection

We concentrated on the impact on the power consumption, particularly as it is affected by the computational time required for feature extraction. This is a crucial step because it prepares the input values that will be fed into the neural network model. By extracting the features, we can reduce the input dimensionality, reducing the size of the input layer and can lead to more efficient processing.

Feature extraction plays a pivotal role in the pre-processing stage of sound classification tasks. Two of the most commonly used features for this purpose are the Short-Time Fourier Transform (STFT) and the Mel-Frequency Cepstral Coefficients (MFCC). Both of these techniques transform raw audio signals into data more manageable and informative for machine learning algorithms.

The Short-Time Fourier Transform (STFT) is computed by splitting the audio signal into fixed-length frames. In our case, we tested different frame sizes in a range between 256 and 1024 samples without overlapping. A Hann windowing function is then applied to each frame to minimize edge effects that can occur during the transformation. This is followed by the computation of the Discrete Fourier Transform (DFT) for each frame, which converts the time-domain signal into the frequency domain using the equation 1.

$$x = \sum_{n=0}^{N-1} x[n] e^{\frac{-j2\pi kn}{N}} \tag{1}$$

The final step in obtaining the spectrogram image involves calculating the magnitude of the imaginary values obtained from the DFT. This results in a vector with a length of $n = \frac{m}{2} + 1$, where m is the window size. This process effectively captures the frequency content of the signal over time, which is essential for many audio analysis tasks. Tab. I reports the accuracy of models that exploits STFT coefficients extracted in *int32*. As previously stated, the models are limited to 2,800 weights. Frame sizes from 256 to 1024 are tested, using only low-frequency coefficients in the model, the same frequency range for the tree cases. From the results, it can be deduced that increasing the precision on the FFT algorithm and utilizing more features will not improve the model scores appreciably.

On the other hand, the Mel-Frequency Cepstral Coefficients (MFCC) are computed through a process that begins similarly to the STFT, but it has been compared the results using two different frame sizes of 512 and 1024 samples without overlapping. After splitting the signal and applying the STFT, a set of triangular filters that are evenly spaced on the Mel scale is applied to the frequency-domain representation. The Mel scale mimics the human ear's perception of sound, focusing more on frequencies that are more perceptible to humans, as described by the following equation:

$$mel(f) = 2595 \cdot log\left(1 + \frac{f}{700}\right) \tag{2}$$

The final step involves applying a Discrete Cosine Transform (DCT) to the filtered signals to obtain a set of coefficients for each frame. The DCT is described by the following two formulas where N is the number of Mel coefficients:

$$X_0 = \frac{1}{\sqrt{N}} \cdot \sum_{n=0}^{N-1} x_n \tag{3}$$

for $k = 0$

$$X_k = \sqrt{\frac{2}{N}} \cdot \sum_{n=0}^{N-1} x_n cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right] \tag{4}$$

for $k = 1, ...N - 1$

FFT	N. of coeff.	N. of weights	Accuracy 3s frame	Accuracy 5s frame
256	70	2817	96.25%	96.33%
512	140	2779	94.69%	96.49%
1024	210	2849	95.39%	95.71%

TABLE I: Accuracy of models that exploit STFT extracted in int32 from 3-second or 5-second frames. The number of coefficients retained is reported.

FFT - MELS	N. of coeff.	N. of weights	Accuracy 3s frame	Accuracy 5s frame
512-10	7	2841	90.09%	90.55%
512-20	17	2785	93.13%	93.52%
512-30	17	2785	94.85%	95.63%
1024-10	7	2841	90.40%	91.26%
1024-20	17	2785	95.55%	95.55%
1024-30	17	2785	95.32%	96.64%

TABLE II: Accuracy of models that exploit MFCCs extracted in int16 from 3-second or 5-second frames. The number of coefficients retained is reported.

These coefficients are then compressed into a vector with a length of n, which corresponds to the number of Mel triangular filters used. This compression retains the most relevant information about the audio signal while reducing the dimensionality, making it suitable for use in neural network models.

As shown in the Tab. II, increasing the number of Mel coefficients improves the accuracy of the models, whereas changing the FFT size does not result in a significant difference. In this case, the extraction is performed in *int16* format, the motivation of this difference respect to STFT it is detailed in IV. Both Tab. II and Tab. I display the accuracy achieved by models utilizing audio of different length. Recording for 5 seconds provides a small improvement in accuracy, however, this scenario requires higher energy consumption.

Both STFT and MFCC are powerful tools in the field of audio classification tasks. They each have their own strengths: STFT provides a comprehensive view of the frequency content over time, while MFCC captures perceptually relevant features that are particularly useful to compress even more the required values that represent the sound. By employing these techniques, we can ensure that the neural network model receives high-quality input data, which is essential for achieving accurate and efficient sound classification.

IV. Results

The extraction process of the audio coefficients is investigated in depth to determine the computational effort required for the microcontroller, as it is one of the most significant portions of the board's total activity. Specifically, we measured the execution time and the current consumption required for extraction on our embedded system. For both metrics, 10 different measurements were performed, and the mean and standard deviation were computed and reported.

As previously stated, the STFT algorithm is easier to compute, and the resultant coefficients are essentially the FFT of the initial audio averaged over time. We noticed that higher precision is required when utilizing the STFT in the machine learning model. This is because high-frequency coefficients have smaller values, and the *int16* format is not sufficient to represent the smaller variance of these coefficients. Therefore, STFT extraction is performed in *int32* format. In this way, the model's metrics achieve results comparable to those of models trained with MFCCs. However, using integers with double precision has the drawback of high flash memory occupation for storing the constant arrays used during the execution of the FFT algorithm, as well as a longer computation time.

MFCC extraction performs additional compression operations, allowing the use of fewer low-frequency coefficients for the model while still achieving good accuracy results. MFCCs exhibits a wider range and variance, for this reason it is possible to use *int16* format, and accordingly, the energy consumption is comparable to STFT extraction. Although the computation is more complex and the current is higher than the STFS, the total time required is shorter. Indeed, STFT extraction takes on average 18% additional time compared to MFCC extraction. This is mainly due to the required higher data precision.

Regarding time measurements, a timer of the microcontroller is employed to count the clock ticks, and the value is finally converted to time. Fig. 2(b) and Fig. 3(b) illustrate the execution time when varying the number of coefficients and the recording time. The extraction time varies because, for a smaller FFT size, the audio is framed in more windows that are shorter, making the computation for a single window faster. This principle applies to both STFT and MFCCs. For the latter, an additional consideration can be made regarding the number of mel-frequencies. Increasing this number results in larger matrices for the computation of the coefficients and the process become longer. For STFT, the shortest time is achieved with a 256-point FFT, while for MFCC extraction, the two sizes tested showed similar values. Increasing the recording duration by 2 seconds results in an additional second required for extraction, whereas increasing the number of MELS has a much smaller impact.

The mean current is estimated by modifying the firmware code of the microcontroller. Measurements are performed using using the Tektronic dm7510 multimeter. The current is acquired continuously, by inserting a delay before and after the audio processing, makes it easy to identify and isolate the current samples corresponding to the execution of the STFT or MFCC algorithm. Fig. 3(a) displays the value obtained for STFT extraction, showing that increasing the FFT size generally decreases the mean current. Conversely, for MFCC extraction, as shown in Fig. 2(a), the mean current increases with a higher number of mel-frequencies, compensating for the longer processing time required. When 5-second frames are used, the current is slightly higher.

The last two graphs, Fig. 2(c) and Fig. 3(c), illustrate the derived energy consumption. The values are obtained by multiplying the mean current with the mean time and the nominal voltage of the supply battery, which is $3.6V$. We noticed

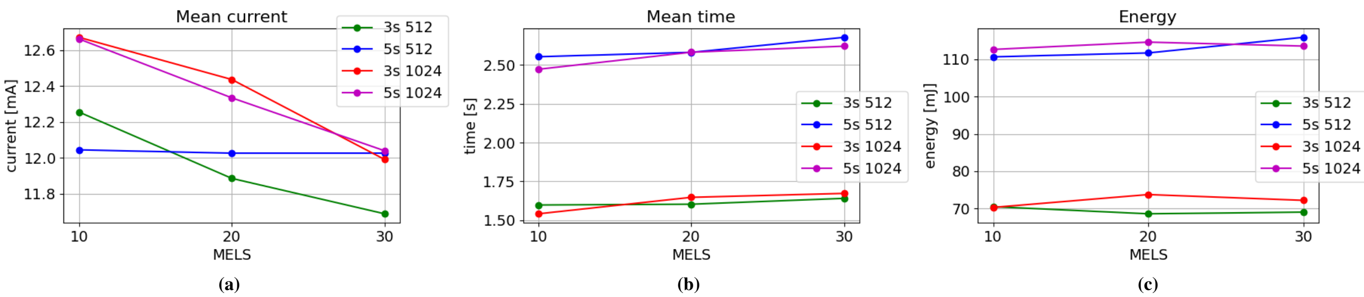

Fig. 2: Measurements on current and time for MFCCs extraction and computed energy. 10 values are measured, mean and standard deviation is computed, results are provided in Tab. III.

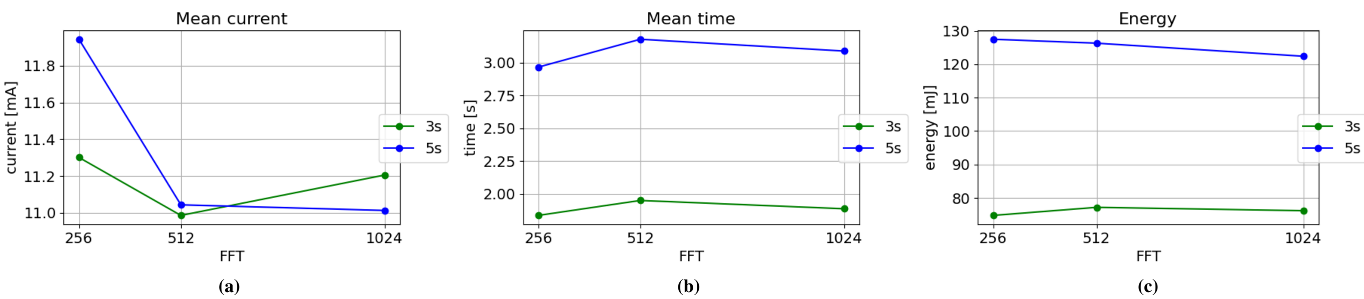

Fig. 3: Measurements on current and time for STFT extraction and computed energy. 10 values are measured, mean and standard deviation is computed, results are provided in Tab. III.

that STFT extraction consumes more energy, approximately 9%, due to the higher processing times, which outweigh the effect of a smaller current. The algorithm's parameter (FFT size and number of mel-frequencies) have a limited influence on the energy consumption, as the effects of higher current and shorter processing times tend to compensate for each other. The most significant increase in energy consumption is observed with the use of longer frame lengths.

These findings highlight the trade-offs between model accuracy, data representation, computational effort, and energy efficiency, providing a methodology for optimizing and choosing parameters for audio processing for feature extraction process.

V. Conclusions

This paper presents a significant investigation into the energy impacts of audio feature extraction for Tiny Machine Learning (TinyML) applications in IoT devices, specifically tailored for monitoring the health of bee colonies. Our study focuses on the extraction of Short-Time Fourier Transform (STFT) and Mel-Frequency Cepstral Coefficients (MFCC), which are critical for predicting the presence of the queen bee, a vital indicator of colony health.

STFT and MFCC require considerable computational resources that, in turn, impact the energy efficiency of the IoT devices used in bee monitoring. MFCC, in particular, while providing better compression and capturing perceptually relevant features, requires a higher computational effort compared to STFT. This underscores a crucial trade-off between computational accuracy and energy efficiency in the design of embedded systems for audio processing.

The deployment of specialized hardware could offer a promising solution to mitigate these challenges. By leveraging hardware acceleration, such as Digital Signal Processors (DSPs) or Field-Programmable Gate Arrays (FPGAs), the feature extraction process can be significantly sped up, reducing the energy consumption and enabling more frequent and reliable monitoring without compromising the battery life of the devices.

In addition, optimizing feature extraction algorithms, data

MFCC				
	3-s audio		5-s audio	
FFT-MELS	Current	Time	Current	Time
512-10	12,255±90	1,598.1±3.1	12.044±43	2.5525±4.8
512-20	11,886±75	1,603.1±2.6	12.026±53	2.5807±5.6
512-30	11,688±99	1,640.8±7.0	12.026±35	2.6773±3.6
1024-10	12,671±68	1,541.3±3.4	12.662±67	2.4717±3.2
1024-20	12,436±51	1,647.4±5.5	12.335±51	2.5815±3.4
1024-30	11,991±77	1,672.7±8.9	12.039±63	2.6204±6.3
STFT				
	3-s audio		5-s audio	
FFT	Current	Time	Current	Time
256	11.301±99	1.8375±2.3	11.943±57	2.9651±3.7
512	10.986±99	1.9511±2.5	11.044±55	3.1770±2.9
1024	11.206±59	1.8881±3.4	11.013±62	3.0872±4.3

TABLE III: Measurements on MFCCs and STFT extraction, the current is expressed in: mean±std [μA], time is expressed in: mean±std [ms].

precision and adjusting the audio processing parameters like the frame size and the number of coefficients can further refine the balance between performance and power usage. Our experimental results suggest that careful selection of these parameters can minimize energy demands while maintaining high accuracy in queen bee detection.

ACKNOWLEDGMENT

This publication is part of the project PNRR-NGEU which has received funding from the MUR – DM 352/2022.

REFERENCES

[1] A.-M. Klein, B. E. Vaissière, J. H. Cane, I. Steffan-Dewenter, S. A. Cunningham, C. Kremen, and T. Tscharntke, "Importance of pollinators in changing landscapes for world crops," *Proceedings of the Royal Society B: Biological Sciences*, vol. 274, no. 1608, pp. 303–313, 2007.

[2] D. VanEngelsdorp, J. Hayes, R. Underwood, and J. Pettis, "A survey of honey bee colony losses in the united states, fall 2008 to spring 2009," *Journal of Apicultural Research*, vol. 49, 01 2010.

[3] 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.

[4] K. Kelly, A. Garavito, and D. vanEngelsdorp, "Population Growth in Honey Bee (Apis Mellifera) Colonies Is Associated with Visitation from Non-Natal Bees," *Scientific Reports*, vol. 11, no. 1, p. 7092, 03 2021. [Online]. Available: https://doi.org/10.1038/s41598-021-86558-8

[5] B. P. Oldroyd, "What's killing american honey bees?" *PLOS Biology*, vol. 5, no. 6, pp. 1–5, 06 2007. [Online]. Available: https://doi.org/10.1371/journal.pbio.0050168

[6] D. Elzinga and C. Strickland, "Generalized stressors on hive and forager bee colonies," *Bulletin of Mathematical Biology*, vol. 85, p. 112, 10 2023.

[7] U. S. D. of Agricolture (USDA). (2023) Honey bee colonies. [Online]. Available: https://downloads.usda.library.cornell.edu/usda-esmis/files/rn301137d/4m90gc28p/gq67m7401/hcny0823.pdf

[8] B. B.-I. Boris Maciejovsky and B. Baer, "Hobbyist beekeepers and their importance for future beekeeping and food production," *Bee World*, vol. 100, no. 3-4, pp. 80–83, 2023.

[9] 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.

[10] T. Cejrowski, J. Szymański, H. Mora, and D. Gil, "Detection of the bee queen presence using sound analysis," in *Intelligent Information and Database Systems*, N. T. Nguyen, D. H. Hoang, T.-P. Hong, H. Pham, and B. Trawiński, Eds. Cham: Springer International Publishing, 2018, pp. 297–306.

[11] 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

[12] A. Zgank, "Bee swarm activity acoustic classification for an iot-based farm service," *Sensors*, vol. 20, no. 1, p. 21, 2019.

[13] 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.

[14] L. Barbisan, G. Turvani, and R. Fabrizio, "Audio-based identification of queen bee presence inside beehives," in *2023 IEEE Conference on AgriFood Electronics (CAFE)*. IEEE, 2023, pp. 70–74.

[15] L. Barbisan, G. Turvani, and F. Riente, "A machine learning approach for queen bee detection through remote audio sensing to safeguard honeybee colonies," *IEEE Transactions on AgriFood Electronics*, pp. 1–8, 2024.

[16] 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, 2021.

[17] A. Robles-Guerrero, T. Saucedo-Anaya, E. González-Ramírez, and J. I. De la Rosa-Vargas, "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*, vol. 159, pp. 69–74, 2019.

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

[19] I. Nolasco, A. Terenzi, S. Cecchi, S. Orcioni, H. L. Bear, and E. Benetos, "Audio-Based identification of Beehive states: The dataset," May 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2667806