

IDENTIFY THE BEEHIVE SOUND USING DEEP LEARNING

¹G.Prabhakar, Associate Professor , CSE, gantelaprabhakar@gmail.com
Swarna Bharathi institute of science and technology,
Khammam

² N.Savitha, Assistant Professor, CSE(DS), savitha.natuva@gmail.com
Swarna Bharathi institute of science and technology,
Khammam

³Ameena nasreen, Assistant Professor, CSE(AIML), amena.nasreen.md@gmail.com
Swarna Bharathi institute of science and technology,
Khammam

ABSTRACT:

Flowers are necessary to purge the dreary air. The life cycle of flowering plants includes many stages, such as pollination, fertilization, blooming, seed development, dispersal, and germination. Approximately 75% of all blooming plants are pollinated by honeybees. Honeybee populations are steadily declining due to threats to their natural habitats from environmental pollution, climate change, the destruction of natural landscapes, and other factors. Consequently, some scholars are working to find a solution. Variations in recordings of beehive noises may be found using acoustic categorization. To distinguish bee sounds from non-beehive noises, we use deep learning techniques—specifically, Sequential Neural Network, Convolutional Neural Network, and Recurrent Neural Network—to the recorded sounds in this study. We also compare the deep learning methods with other well-known non-deep learning methods, such as Random Forest, Naïve Bayes, Support Vector Machine, and Decision Tree. The whole set of recorded sounds (25–75%) is also used to verify the procedures.

OVERVIEW

Monitoring honeybee colonies and determining their whereabouts is one of the most important research projects for guaranteeing the long-term viability of

natural habitats. As a result, many researchers like working in that subject. Honeybee colonies and beehive noises may be effectively identified using the audio-based method [1–3]. Creating systems that can discriminate between bee and non-bee noises is the first stage in creating audio-based beehive monitoring devices. The events and environment around the hive are often linked to noises coming from non-bee sources, such as cities, animals, rain, and maintenance. As a result, our study automatically detects sound recordings from beehives. Due to their possible shorter length, beehive sounds are successfully separated from the collection of noises from non-beehives.

We initially preprocess all of the two-second sound waves originating from beehives in order to differentiate between bee and non-bee sounds. Following that, 134 features are extracted from the waves, including Root-Mean-Square (RMSE) Energy, Spectral Centroid, Spectral Bandwidth, Spectral Rolloff, Zero Crossing Rate, Mel Frequency Cepstral Coefficients (MFCCs), which are made up of 128 distinct sequences (mfcc1 to mfcc128), and Chromagram STFT (which provides information about the pitch category and signal pattern) (International Journal of Computer Science & Information Technology (IJCSIT) Vol 14, No 4, August 2022 14). We choose 26 of the 134 most valuable qualities using feature selection techniques like Select_K_Best and Correlation Coefficient. Both deep learning and non-deep learning classification techniques are trained on these 26 feature-based

datasets. The testing accuracy of deep learning methods ranges from 85.04 to 99.26%, while non-deep learning approaches range from 88.06 to 97.74%. To help with this inquiry, five distinct sounds—25–75% of beehive sounds and non-beehive sounds—are also chosen at random from the raw wave file. This section provides a quick overview of the planned study on beehive sound recognition. The relevant study on the recognition and categorization of bee sounds is presented in Section 2. A thorough description of the datasets, feature extraction processes, and their application to various machine learning approaches are given in Section 3, which also covers the methodology of the proposed research. The outcomes of validation and classification techniques using integrated recorded datasets are covered in Section 4. Limitations and further work are mentioned in Section 5. In section 6, we complete this assignment before moving on.

2. CONNECTED WORK

Given that a beehive's sound is a combination of the unique sounds produced by each colony, we chose to investigate the frequency domain recording of the sounds in our dataset. Mel-frequency Cepstral Coefficients (MFCCs) are traits that researchers utilized to distinguish noises coming from beehives [4]. To detect beehive sounds, denoising methods and practical components such as spectral contrast, chroma, Wavelet transform, and features were also used. Furthermore, by using data preprocessing, manually created features, and subject expertise, the beehive sound analysis study removes non-beehive noises from the recordings. To exclude non-bee noises that were not expected to be part of the bee sound class, the authors of [5] used a Butterworth filter with cut-off frequencies of 100 Hz and 2000 Hz to filter the acoustic data. Three categories of sounds—ambient, beehive, and cricket—are distinguished by the study [6]. The amount of domain knowledge and handcrafted characteristics provided by bias may be diminished by machine learning methodologies, especially deep learning techniques. Additionally, the modeling capabilities of sound recognition systems may be limited. A notable method to record that sound through hand-crafted attributes [10] could be the IoT-based bee swarm activity acoustic classification technique [9] or the normal acoustic classification technique [8], as the paper [7] showed that DNNs outperformed shallower approaches like Gaussian mixture models in environmental sound scene analysis (GMMs). To construct an annotated dataset, the authors of [11] used sound recordings from beehives. Machine learning techniques like as SVM [12,13] and CNN [14,15] were thought to be excellent sound

identification systems. Additionally, a detailed comparison of CNN and machine learning classification techniques' effects on beehive sound detection is available in [16]. However, both beehive sound identification utilizing data gathered by IoT sensors [18] and beehive monitoring [17] favor machine learning classification techniques. While MFCC [20, 21] is the most popular feature extraction technique, Random Forest (RF) and Support Vector Machines (SVM) [19] are the most popular non-deep learning classifiers for detecting bee buzzing sounds. Moreover, two efficient methods for forecasting bee sound from flying sound are Decision Tree (J48) and Naïve Bayes [22]. August 15th, 2022: International Journal of Computer Science & Information Technology (IJCSIT), Volume 14, Issue 4. Up until now, the study that has been previously detailed has focused on feature engineering, which includes feature extraction and the process of choosing the optimal qualities for classification that are lacking. It appears too important to determine the practical aspects for beehive sound identification since so many components may be retrieved from the beehive audio. Furthermore, it is evident that feature engineering is necessary to identify the necessary characteristics in order to make them compatible with the methods used to get noteworthy results.

TECHNIQUES

Using a bespoke framework, this study developed a model to categorize bee sounds from internet datasets and non-beehive noises using both deep learning and non-deep learning techniques (machine learning algorithms). The framework is divided into four phases (Figure 1), the first of which outlines the information gathered online. The next step demonstrates how the datasets were preprocessed using a few steps. Following preprocessing, the dataset's available features are extracted, and the best features are chosen using feature extraction and feature selection. The stage is for putting the desired machine learning methods into practice. The approach used in this study is shown in Figure 1.

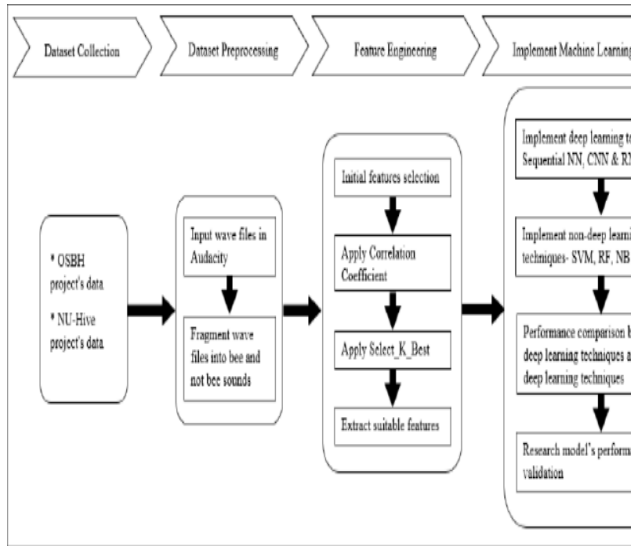


Figure 1. Methodology of this research

3.1 Gathering Datasets

It goes without saying that gathering data from reputable sources and doing research is essential to the development of this research framework.

Consequently, we get datasets from two initiatives: the NU-Hive and the Open-Source Beehive (OSBH) [11]. These projects' main goal is to create beehive monitoring systems that can recognize and forecast the condition of hives.

3.2. Preprocessing the Dataset

The Sonic Visualiser is used in the previous study [11] to complete the sound annotation tasks. The entire recording was separated into Bee and noBee intervals and saved as text files by labeling it with these pairs of instances that corresponded to the beginning and ending of external sound periods (International Journal of Computer Science & Information Technology (IJCSIT) Vol 14, No 4, August 2022 16. The periods when an external sound is heard are indicated by the noBee intervals. Figure 2 provides an illustration of this procedure.

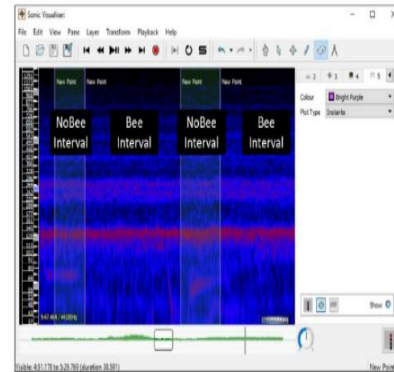


Figure 2. Annotation procedure in the earlier research for one audio file by Sonic Visualiser3

We chose to use the Audacity program for this study in order to navigate through the wave files (in the dataset), which are divided into sounds that are bee-related and noises that are not. Divide them into two-second wave files (bee or not bee) using the annotation provided. A total of 4070 wave files including 1100 and 2970 bee and non-bee sound samples are recovered. Figure 3 illustrates a waveform example utilizing this technique.

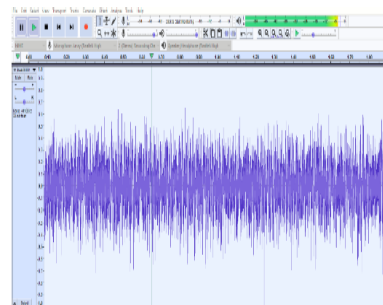


Figure 3. Annotation procedure in the current research for one audio file by Audacity

The audio files are split into blocks of two seconds and processed at a sample rate of 22050 Hz. The audio signal is repeated until the necessary block length is reached for segments whose lengths are less than the designated block length. Each block is assigned a label based on the annotations that are currently in place. The label Bee is applied if there are no external mixed sound intervals during the whole section. Likewise, the label noBee is used if an external sound event occurs in at least part of the segment. A sample amplitude from the combined form of the Bee and noBee sound is shown in Figure 4. International Journal of Computer Science & Information Technology (IJCSIT)

3.3. Engineering Features

Features are useful attributes for precisely describing a large collection of data. To begin building a machine learning approach, we must first extract the appropriate features from the data set and then follow a process to select the most relevant, consistent, and non-redundant features. Feature extraction and appropriate feature selection are the two naming phases that comprise feature engineering.

3.3.1. Extraction of Initial Features

At the first stage of this feature engineering method, 134 features are retrieved, which is a significant amount. Spectral Centroid, Spectral Bandwidth, Chromagram, Zero Crossing Rate, Roll Off, Root Mean Square Error, and Mel Frequency Cepstral Coefficients (of 128 distinct sequences) are all included in this extensive feature set. The "center of mass" of a sound is its spectral centroid, which is calculated by averaging the weight of the sound frequencies (see Equation 1). The Spectral Centroid would have been identified at the middle of a sound if its frequencies were constant throughout; on the other hand, if the sound's ending section had high frequencies, the centroid may have been found close to its end. Spectral shape is measured by the centroid, and "brighter" timbral textures with a higher percentage of high frequencies are associated with relatively high centroid values. $M[n]$ represents the Fourier transform value at time frame t and frequency bin n for the purposes of this equation.

$$C t = \sum n M [n] / \sum M t [n]$$

The Spectral Bandwidth is used to compute the Spectral Bandwidth: $N n = 1 N n = 1(1)$. It gives details on the variance of the audio signal in relation to the Spectral Centroid. Regarding perceived timbre, there is a high correlation between the spectral bandwidth and perceived timbre. The quantity of energy dispersed over various frequency bands determines the bandwidth. In terms of mathematics, it is the average weight of the separations between the Spectral Centroid and the frequency bands.

Chromagram STFT to get the chroma characteristics in order to represent the pitch data of the signals. The main idea behind chroma features is to compile all of the spectral data related to a certain pitch class into a single coefficient. The audio stream may be divided into 12 distinct pitches based on its audio octave. The whole spectrum is projected into 12 pitches via chromagram. A signal that deviates from the horizontal axis's polar location is detected by the Zero Crossing Rate. The mean and variance of zero crossings throughout the period in the texture window are employed as features, and the information offered by time-domain zero-crossings is used to measure the signal's noisiness. This metric measures a signal's

capacity to change from positive to zero to negative and from negative to zero to positive within a certain time frame. Equation 2 illustrates the equation for Zero Crossing Rate across a certain time domain. It is equal to

$$Z t 1/2 \sum |s i (x [n] - s i g n (x [n - 1]))| N n = 1 (2)$$

International Journal of Computer Science & Information Technology (IJCSIT)

In a frequency response curve, roll off shows the boundaries of the deep bass and high treble. Additionally, 85% of the magnitude distribution was centered at the frequency $R t$, which indicates the Roll Off: $\sum M [n] R t n = 1 = 0.85 \sum M t [R t n = 1 n]$.

The variation and mean of the Roll Off throughout the time-frames within the texture window are taken to build many features. Additionally, RMSE is a noteworthy characteristic for study on audio signal categorization. As a feature, it shows the wave signal's Root-Mean-Square energy value. Continuously fluctuating audio signals are averaged using the Root Mean Square Error. Its value is determined by calculating the audio signal frame-by-frame.

4. OUTCOME :

The performance analysis of the methods used and the desired qualities for this study are presented in this part. The various methods, features, input layers, activation functions, and optimizers all have distinct performances. The most appropriate method and practical aspects are found for this study via output observation.

Table 3. Performance of Sequential NN based on different activation functions and optimizers.

Optimizer	Adam	Ada Delta	Ada Grad	Ada Max	FTRL	Nadam	RMS prop	SGD
Relu	98.40%	93.73%	98.16%	98.03%	71.74%	98.52%	99.01%	98.03%
Sigmoid	97.79%	72.72%	89.56%	99.26%	75.68%	98.03%	98.16%	90.42%
Tanh	97.79%	93.37%	96.68%	98.28%	85.14%	97.79%	98.53%	98.16%

Initially, we must determine the optimal Sequential NN design. Python's Keras library has three activation functions: Tanh (Hyperbolic Tangent Activation Function), Sigmoid, and Relu (Rectified Linear Unit). Additionally, Keras has eight optimizers: Adam (Adaptive Moment Estimation), AdaDelta, Ada-Grad, AdaMax, FTRL (Follow the Regularized Leader), Nadam (Nesterov-accelerated Adaptive Moment Estimation), RMSProp (Root Mean Squared Propagation), and SGD (Stochastic Gradient Descent).

For performance analysis, this study must choose the optimal activation function and optimizer from the list of activation functions and optimizers given above. Table 3 shows that the combination of the AdaMax optimizer with the Sigmoid activation function yields the best performance for Sequential NN, with an accuracy of 99.26%. The FTRL activation function with Relu optimizer provide the lowest accuracy, 71.74%. notable results utilizing the AdaMax optimizer with the activator functions Tanh (98.28% accuracy) and Relu (98.03% accuracy). Given that the optimizer, AdaMax, and activation function, Sigmoid, have shown the greatest performance, it is advised that all deep learning approaches be used in conjunction going forward.

FINAL RESULTS

In this study, we use the annotated datasets from previous studies to construct our own datasets. In the first stage, 134 characteristics are chosen in total. Two feature engineering techniques are used to choose the 26 best features: Correlation Coefficient and Select_K_Best. Using the 26 feature datasets, we investigate both deep learning and non-deep learning machine learning techniques and evaluate their accuracy. With an accuracy of 99.28%, sequential neural networks with sigmoid activation functions and AdaMax optimizers outperform the other deep learning methods. With an accuracy of 97.74%, Random Forest outperforms the other non-deep learning methods. In the combined data, sequential neural networks are likewise exhibiting the greatest performance. It is advised to employ the model going forward since it operates correctly with up to 25% of the total noises. The study also reveals that RNN outperforms CNN in distinguishing bee sounds from non-beehive noises, but falls short of Sequential NN. Since CNN works well in picture datasets, RNN also demonstrates good performance in time series, text, or audio datasets, and Sequential NN performs well in tabular type datasets, this is a typical situation. The audio stream will be sent straight to RNN in our next study, and CNN will evaluate their performance.

REFERENCES

[1] In Computational Analysis of Sound Scenes and Events, Cham: Springer International Publishing, 2018, pp. 303–333, D. Stowell presents "Computational Bioacoustic Scene Analysis."
[2] "Identification of the honey bee swarming process by analysing the time course of hive vibrations," Comput. Electron. Agric., vol. 76, no. 1, pp. 44–50, 2011, M. Bencsik, J. Bencsik, M. Baxter, A. Lucian, J.

Romieu, and M. Millet. A. Zacepins, A. Kviessis, E. Stalidzans, M. Liepniece, and J. Meitalovs,
[3] "Single-point temperature monitoring for remote detection of honey bee colony swarming," Biosyst. Eng., vol. 148, pp. 76–80, 2016.
[4] C. E. Galván-Tejada, T. Saucedo-Anaya, E. González-Ramírez, and A. Robles-Guerrero, "A preliminary study on the use of honey bee buzz frequency analysis for automatic health status recognition," Research in Computing Science, vol. 142, no. 1, pp. 89–98, 2017.
[5] A study by S. Ferrari, M. Silva, M. Guarino, and D. Berckmans titled "Swarming sound monitoring in bee hives for early detection of the swarming period" was published in 2008.
[6] P. Amlathe, "Normative machine learning methods in Audio Beehive Monitoring: Using logistic regression, K-nearest neighbor, random forest, and support vector machine to classify audio samples," <https://digitalcommons.usu.edu/etd/7050/>, DigitalCommons@USU.
[7] "A comparison of Deep Learning methods for environmental sound detection," presented by J. Li, W. Dai, F. Metze, S. Qu, and S. Das at the 2017 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP). Vol. 14, No. 4, August 2022, International Journal of Computer Science & Information Technology (IJCSIT)
[8] "Acoustic features for environmental sound analysis," by R. Serizel, V. Bisot, S. Essid, and G. Richard, in Computational Analysis of Sound Scenes and Events, Cham: Springer International Publishing, 2018, pp. 71–101.
[9] A. Zgank, "Deep neural networks for IoT-based bee swarm activity acoustic classification," Sensors (Basel), vol. 21, no. 3, p. 676, 2021.
[10] "Classification and visualization of beehive sounds using machine learning algorithms and Grad-CAM in acoustic scenes,
[11]" Math. Probl. Eng., vol. 2021, pp. 1–13, 2021, J. Kim, J. Oh, and T.-Y. Heo. The study "To bee or not to bee: Investigating machine learning approaches for beehive sound recognition" was published in 2018 by I. Nolasco and E. Benetos.