AI-Based Acoustic Analysis for Health Classification of Beehives Using Machine Learning

YALLA LILLY GRACE

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

A. Naga Raju

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

The health of honeybee colonies plays a critical role in global agriculture due to their importance in pollination and ecosystem stability. However, traditional methods of monitoring beehive health are labor-intensive, intrusive, and require expert knowledge. This project presents an intelligent, non-invasive system for classifying the health status of beehives using acoustic signals and machine learning techniques. The proposed system captures audio signals generated within the hive and analyzes them to determine whether the colony is in a healthy or stressed state.The system employs advanced audio signal processing techniques to extract meaningful features from hive sounds. A total of 17 features are derived, including Mel-Frequency Cepstral Coefficients (MFCCs), spectral centroid, spectral bandwidth, spectral rolloff, and zero-crossing rate. These features capture both time-domain and frequency-domain characteristics of the audio signals, providing a comprehensive representation of hive activity.A Random Forest Classifier is used to train the model on labeled datasets of beehive sounds. The classifier learns patterns associated with different hive conditions, such as normal activity, swarming, or stress due to environmental factors or disease. Once trained, the model can accurately classify new audio inputs, providing real-time insights into hive health.The system is implemented with a user-friendly graphical interface using Python's Tkinter library. Users can upload audio files, view classification results, and visualize signal characteristics through FFT plots, MFCC spectrograms, and spectral feature graphs. These visualizations enhance interpretability and provide deeper insights into the acoustic patterns of beehives.Experimental results demonstrate that the proposed system achieves high accuracy in classifying hive health conditions. The use of machine learning significantly reduces human intervention and enables continuous monitoring. This approach is scalable and can be integrated with IoT devices for real-time field deployment.Overall, this project contributes to precision apiculture by providing an efficient, cost-effective, and non-invasive solution for monitoring beehive health. It has the potential to improve honey production, reduce colony losses, and support sustainable agricultural practices.

Keywords: Beehive Monitoring, Acoustic Signal Processing, Machine Learning, Random Forest, MFCC, Smart Agriculture, Precision Apiculture, Audio Classification, IoT in Agriculture

I. INTRODUCTION

Honeybees are tiny engineers running one of nature's most sophisticated micro-cities. Inside a hive, thousands of bees communicate, coordinate, and survive through a complex system of vibrations and sounds. These acoustic signals carry valuable information about the colony's health and behavior. Monitoring these signals offers a promising approach to understanding and maintaining hive health without disturbing the bees.Traditional beekeeping methods rely on manual inspection, which involves opening the hive and physically examining the colony. This process is time-consuming, requires expertise, and can stress the bees, potentially disrupting their natural activities. Moreover, early signs of disease or stress are often subtle and may go unnoticed until significant damage has occurred.

With the advancement of artificial intelligence and signal processing techniques, it is now possible to analyze audio signals for meaningful patterns. Acoustic monitoring provides a non-invasive and continuous method for assessing hive conditions. By capturing the natural sounds produced by bees, such as buzzing frequencies and vibration patterns, researchers can identify indicators of health, stress, or abnormal behavior.This project focuses on developing a machine learning-based system to classify beehive health using audio signals. The system utilizes feature extraction techniques to convert raw audio data into numerical representations that can be processed by a classifier. Among these features, MFCCs are particularly effective as they capture perceptually relevant aspects of sound, making them widely used in speech and audio recognition applications.The Random Forest algorithm is chosen for its robustness, high accuracy, and ability to handle complex datasets. It works by constructing multiple decision trees and combining their outputs to produce a final classification. This ensemble approach reduces overfitting and improves generalization performance.In addition to classification, the system provides visualization tools that allow users to explore the characteristics of the audio signals. FFT plots reveal frequency components, MFCC spectrograms display time-frequency variations, and spectral feature graphs show trends in acoustic properties. These tools enhance understanding and support decision-making.The integration of machine learning with acoustic monitoring represents a significant step toward smart beekeeping. This approach not only improves efficiency but also enables early detection of potential issues, reducing colony losses and increasing productivity. The proposed system demonstrates how technology can be leveraged to support sustainable agriculture and environmental conservation.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

Recent research has explored various techniques for monitoring beehive health, ranging from sensor-based systems to image processing and acoustic analysis. Among these, acoustic monitoring has gained significant attention due to its non-invasive nature and ability to capture real-time hive activity.Early studies focused on analyzing the frequency spectrum of bee sounds to identify patterns associated with swarming and queen absence. Researchers observed that healthy colonies exhibit stable buzzing frequencies, while stressed colonies show irregular patterns. These findings laid the foundation for using audio signals as indicators of hive health.Traditional methods employed basic signal processing techniques such as Fourier Transform to analyze frequency components. While effective, these methods lacked the ability to capture complex patterns in the data. With the emergence of machine learning, researchers began using classification algorithms such as Support Vector Machines (SVM), k-Nearest Neighbors (KNN), and Artificial Neural Networks (ANN) to improve accuracy.MFCC-based feature extraction became widely adopted due to its effectiveness in representing audio signals. Studies demonstrated that MFCC features, combined with machine learning classifiers, significantly improved classification performance. Deep learning approaches, including Convolutional Neural Networks (CNNs), have also been used to automatically extract features from spectrograms.Several IoT-based systems have been developed for real-time hive monitoring. These systems use microphones and sensors to collect data and transmit it to cloud platforms for analysis. While effective, they often require high computational resources and infrastructure.Random Forest classifiers have been used in various environmental monitoring applications due to their robustness and ability to handle noisy data. Compared to other algorithms, Random Forest provides better interpretability and requires less parameter tuning.Despite these advancements, existing systems face challenges such as high cost, complexity, and lack of user-friendly interfaces. Many solutions are designed for research purposes and are not easily accessible to beekeepers.The proposed system addresses these limitations by combining efficient feature extraction, a robust classification algorithm, and a simple graphical interface. It provides a practical and scalable solution for real-world applications.

III. EXISTING SYSTEM

Existing beehive monitoring systems primarily rely on manual inspection or basic sensor-based approaches. In traditional beekeeping, hive health is assessed by physically opening the hive and observing factors such as bee activity, brood patterns, and honey production. This method is labor-intensive and can disturb the bees, leading to stress and reduced productivity.Some modern systems use environmental sensors to monitor temperature, humidity, and hive weight. While these parameters provide useful information, they do not directly capture behavioral patterns of bees. As a result, these systems may fail to detect early signs of disease or stress.Acoustic-based systems have been introduced to address these limitations. However, many of these systems use simple signal processing techniques and lack advanced classification capabilities. They often rely on threshold-based methods, which are not robust to variations in environmental conditions.Additionally, existing solutions may require expensive hardware and complex setups, making them inaccessible to small-scale beekeepers. The absence of intuitive interfaces further limits their usability.

IV. PROPOSED METHOD

The proposed system introduces an intelligent and user-friendly approach to beehive health classification using acoustic signals and machine learning. It leverages advanced feature extraction techniques to analyze audio data and identify patterns associated with different hive conditions.The system extracts 17 features from audio signals, including

MFCCs and spectral characteristics. These features are used to train a Random Forest classifier, which learns to distinguish between healthy and unhealthy hive states. The trained model is then used to classify new audio inputs with high accuracy.A graphical user interface is developed using Tkinter, allowing users to easily upload audio files and view results. The system also provides visualization tools such as FFT plots, MFCC spectrograms, and spectral feature graphs, enabling users to understand the underlying patterns in the data.

Unlike existing systems, the proposed solution is cost-effective, non-invasive, and easy to use. It does not require specialized hardware and can be deployed on standard computing devices. The system can also be integrated with IoT platforms for real-time monitoring.By combining machine learning with acoustic analysis, the proposed system offers a powerful tool for precision apiculture, improving hive management and supporting sustainable agriculture.

V. IMPLEMENTATION

The implementation of the beehive health classification system is carried out using Python, integrating multiple libraries for audio processing, machine learning, visualization, and graphical user interface development. The system is designed in a modular fashion, ensuring scalability, maintainability, and ease of use.The first phase of implementation involves audio data acquisition and preprocessing. Audio files in formats such as WAV and MP3 are loaded using the librosa library. The audio signals are converted into mono format and normalized to ensure consistency across different samples. This preprocessing step is crucial for eliminating noise and ensuring that the extracted features accurately represent the underlying sound patterns of the beehive.Feature extraction forms the core of the system. A total of 17 features are extracted from each audio file. These include 13 Mel-Frequency Cepstral Coefficients (MFCCs), which capture the perceptual characteristics of sound, along with four spectral features: spectral centroid, spectral bandwidth, spectral rolloff, and zero-crossing rate. These features are combined into a single feature vector using NumPy, providing a compact yet informative representation of the audio signal.The dataset is then split into training and testing sets using the train_test_split function from Scikit-learn. A Random Forest Classifier is employed for model training due to its robustness and ability to handle high-dimensional data. The classifier constructs multiple decision trees and aggregates their outputs to produce accurate predictions. The trained model is saved using the joblib library, enabling reuse without retraining.The classification process involves loading the saved model and applying it to new audio inputs. The extracted features from the input file are passed to the model, which predicts the health status of the beehive. The result is displayed to the user through a graphical interface.The system also includes visualization modules for better interpretability. The Fast Fourier Transform (FFT) is used to analyze the frequency components of the audio signal. MFCC spectrograms provide a time-frequency representation, while spectral feature plots illustrate variations in acoustic properties over time.A user-friendly GUI is developed using Tkinter. The interface includes buttons for selecting audio files, viewing classification results, and generating visualizations. The design ensures that even users with minimal technical knowledge can operate the system effectively.Overall, the implementation integrates audio signal processing, machine learning, and visualization into a cohesive system, providing an efficient and practical solution for beehive health monitoring.

VI. ALGORITHMS

The proposed system utilizes a combination of signal processing and machine learning algorithms to classify beehive health based on acoustic data.The first step involves feature extraction using Mel-Frequency Cepstral Coefficients (MFCCs). MFCCs are derived by applying a Fourier Transform to the audio signal, followed by mapping the frequencies onto a mel scale, taking the logarithm, and performing a Discrete Cosine Transform (DCT). This process captures the perceptual characteristics of sound and reduces dimensionality.In addition to MFCCs, spectral features such as centroid, bandwidth, rolloff, and zero-crossing rate are computed. These features provide insights into the distribution and variation of frequencies within the signal.The classification algorithm used is the Random Forest Classifier. It is an ensemble learning method that constructs multiple decision trees during training. Each tree is built using a random subset of the data and features. The final prediction is obtained by aggregating the outputs of all trees through majority voting.

Algorithm Steps:

1. Load and preprocess audio data
2. Extract MFCC and spectral features
3. Form feature vectors
4. Split dataset into training and testing sets
5. Train Random Forest model
6. Save trained model
7. Load model for prediction
8. Classify new audio input

The Random Forest algorithm is chosen due to its advantages, including high accuracy, resistance to overfitting, and ability to handle noisy data. It also provides feature importance metrics, which help in understanding the contribution of each feature.

VII. SYSTEM DESIGN

The system design of the beehive health classification model follows a layered architecture, ensuring efficient data flow and modular functionality. The design consists of four main components: data acquisition, processing, classification, and user interface.The data acquisition layer is responsible for collecting audio signals from beehives. In the current implementation, audio files are manually uploaded by the user. However, the system can be extended to integrate with IoT devices such as microphones placed inside or near the hive for real-time data collection.

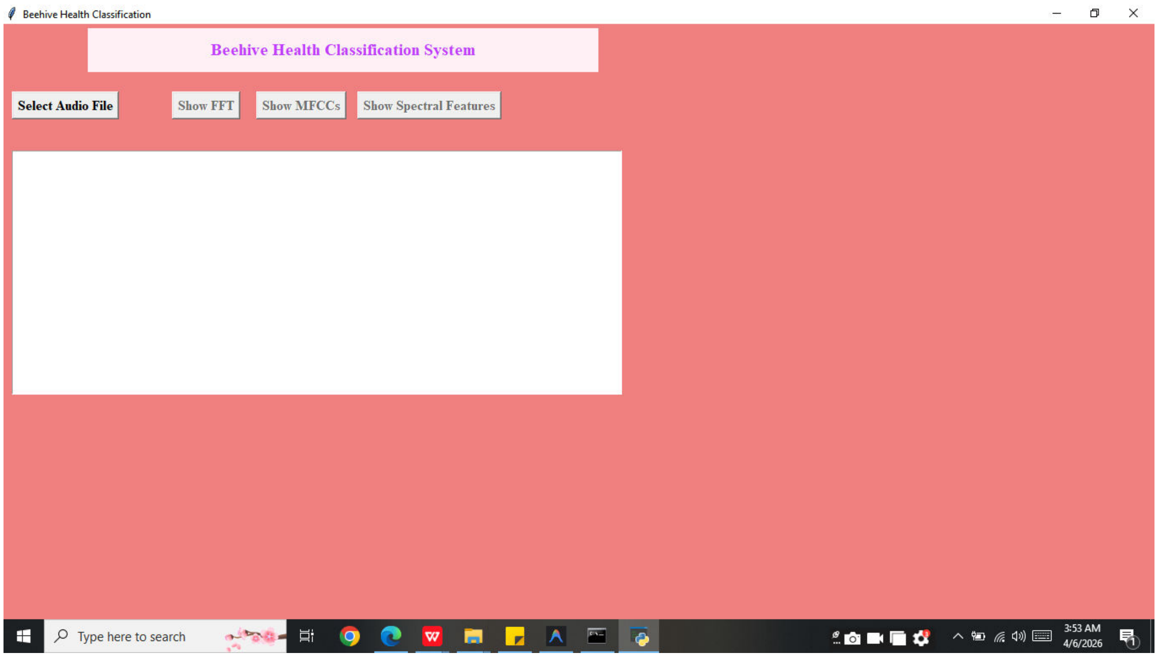
The processing layer handles preprocessing and feature extraction. Once an audio file is selected, it is processed using the librosa library to extract MFCC and spectral features. This layer transforms raw audio signals into structured numerical data that can be used for machine learning.The classification layer is the core of the system. It consists of a trained Random Forest model that takes feature vectors as input and outputs the predicted health status of the beehive. The model is trained offline and stored using joblib, allowing fast and efficient predictions during runtime.The visualization module is integrated within the processing layer. It provides graphical representations of audio features, including FFT plots, MFCC spectrograms, and spectral feature graphs. These visualizations help users understand the characteristics of the audio signals and validate the classification results.The user interface layer is developed using Tkinter. It acts as the interaction point between the user and the system. The interface includes buttons for uploading audio files, initiating classification, and viewing visualizations. A text area is provided to display results and messages.

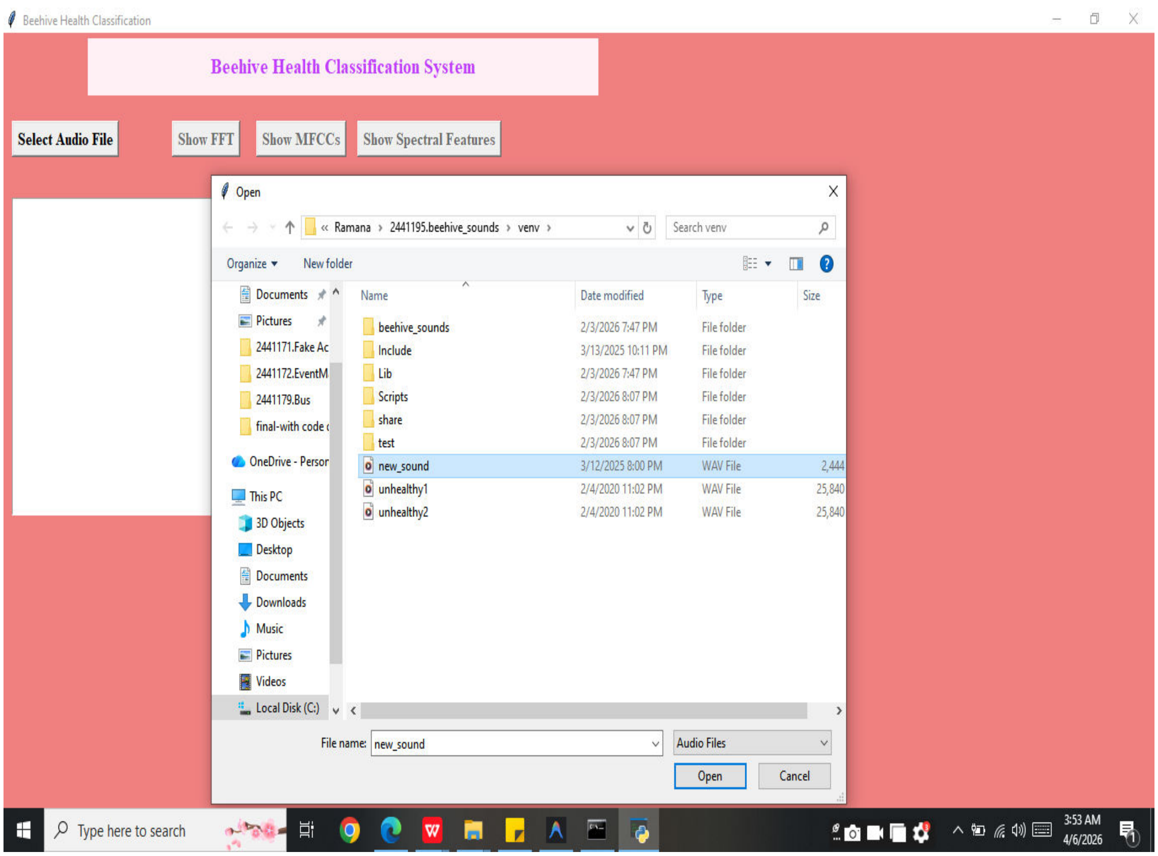
The system follows a sequential workflow:

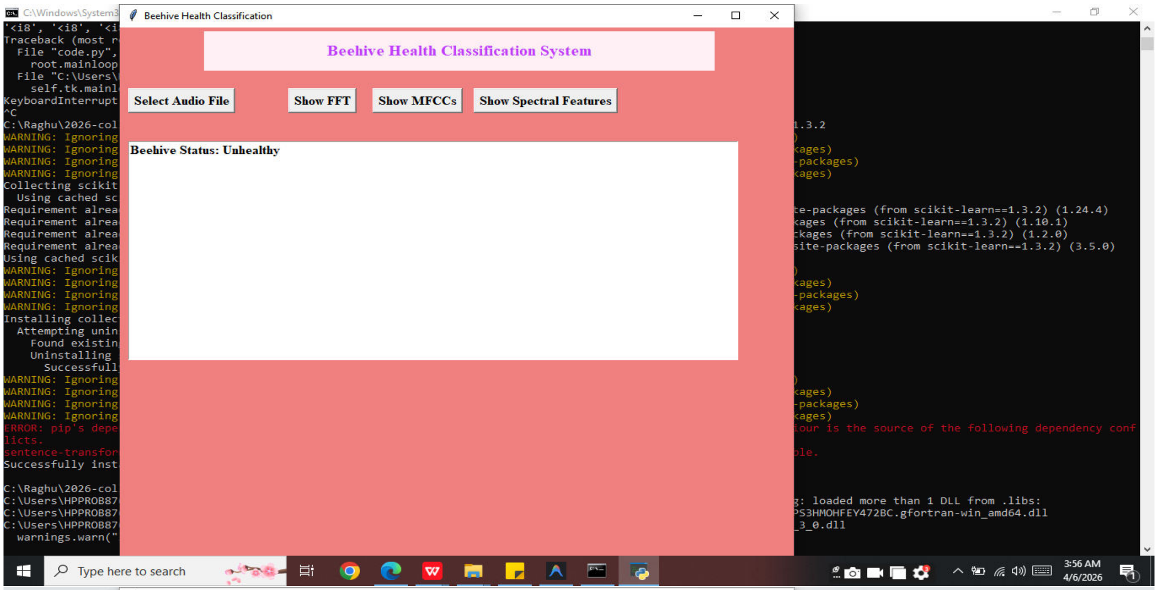
1. User uploads an audio file
2. System extracts features
3. Model predicts hive health
4. Result is displayed
5. User can visualize signal properties

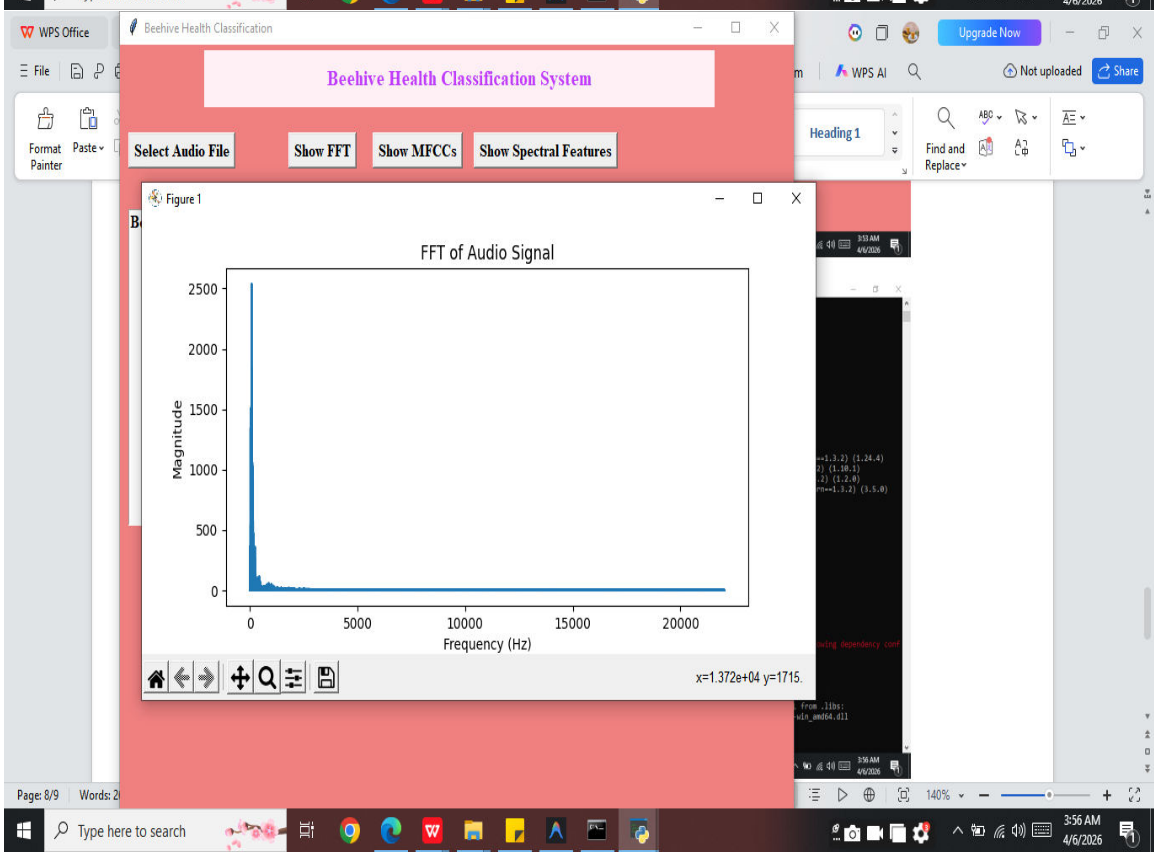
The design ensures modularity, allowing each component to be updated independently. For example, the classification model can be replaced with a deep learning model without affecting other components.Overall, the system design emphasizes usability, scalability, and efficiency, making it suitable for real-world deployment in smart agriculture.

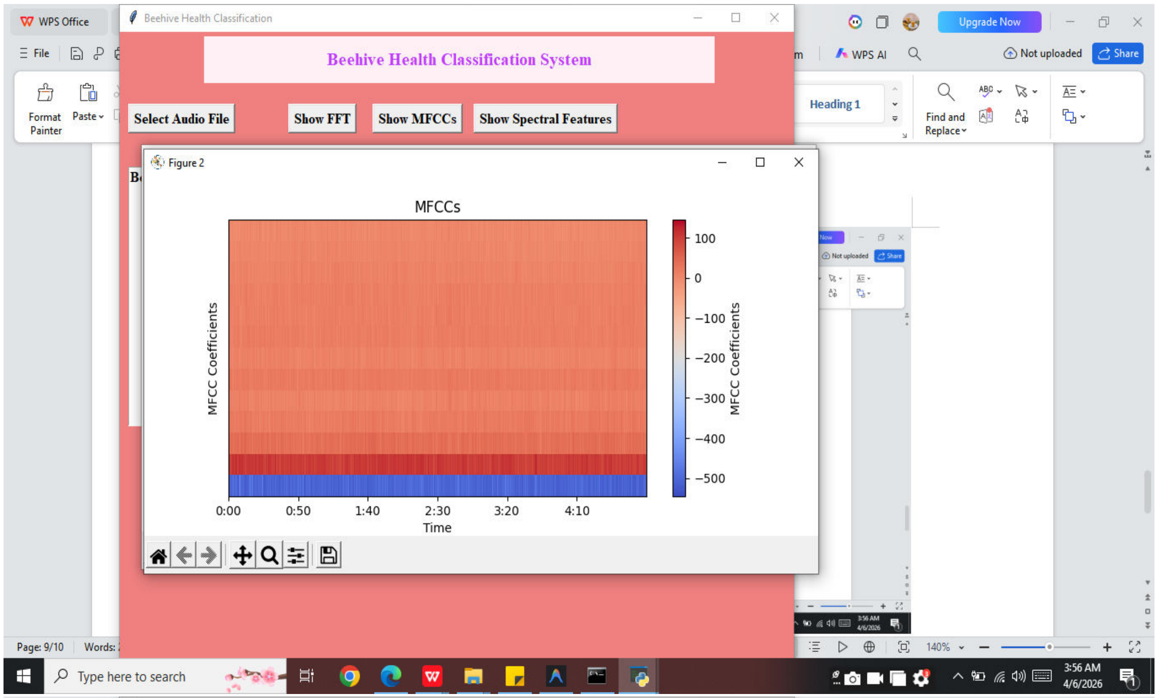
SYSTEM DESIGN IMAGES

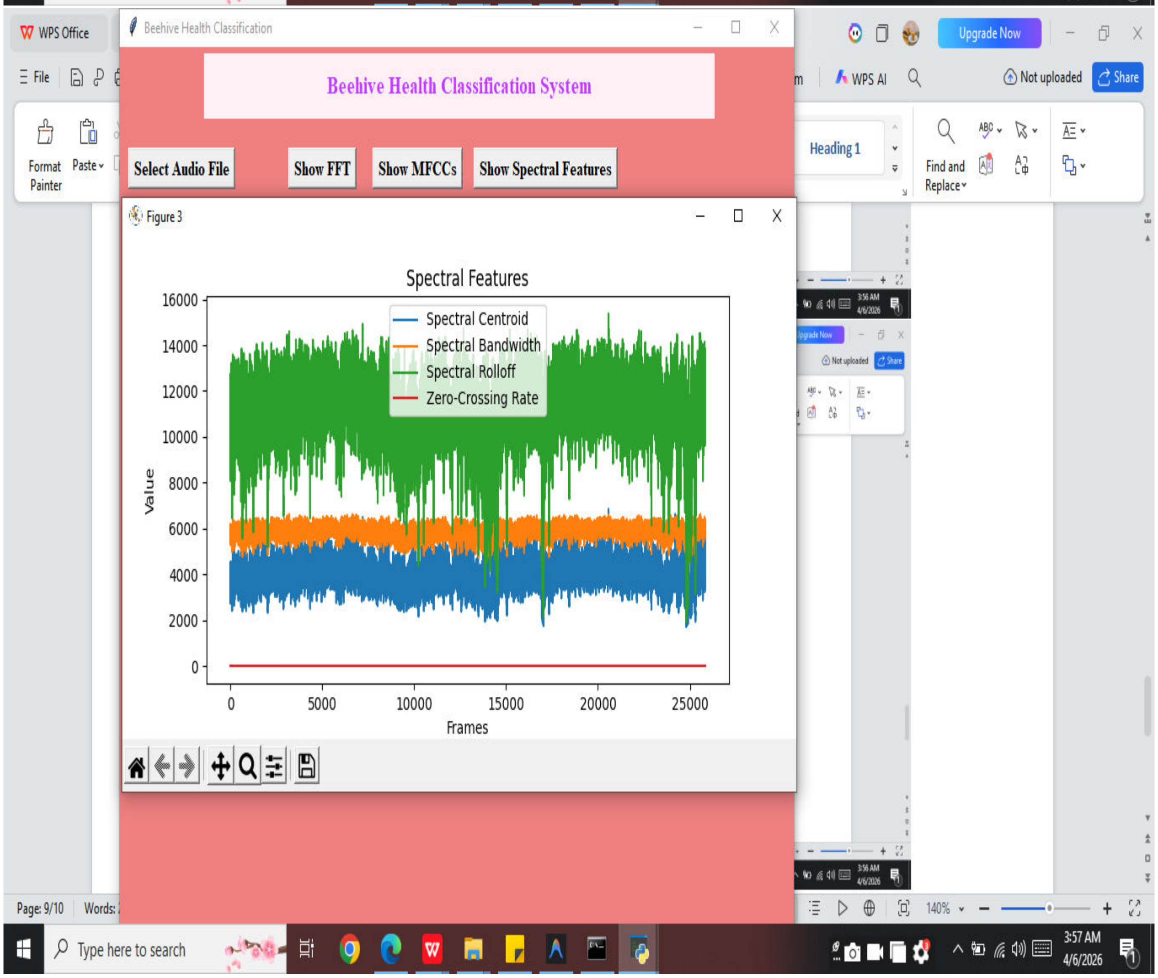


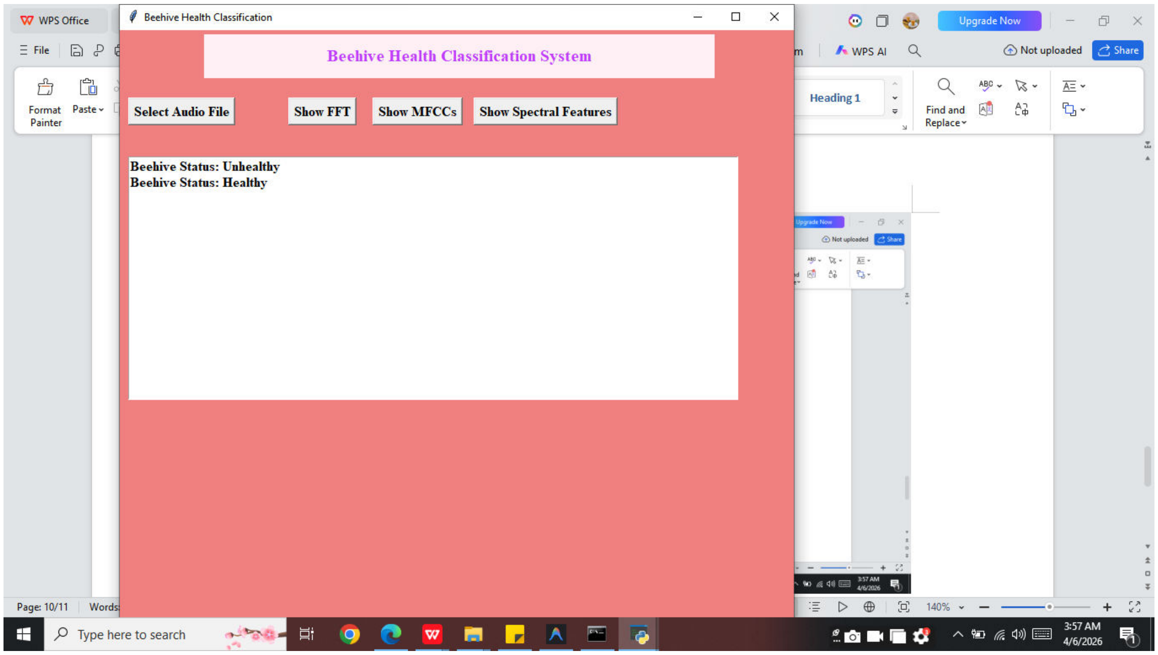



Beehive Health Classification
Beehive Health Classification System
Select Audio File
Show FFT
Show MFCCs
Show Spectral Features
Beehive Status: Unhealthy
Type here to search
3:56 AM
4/6/2026


WPS Office
Beehive Health Classification
Beehive Health Classification System
Select Audio File
Show FFT
Show MFCCs
Show Spectral Features
Figure 1
FFT of Audio Signal
Magnitude
Frequency (Hz)
0
500
1000
1500
2000
2500
0
5000
10000
15000
20000
x=1.372e+04 y=1715.
Upgrade Now
Not uploaded
Share
Heading 1
Find and Replace
Format Painter
Paste
Page: 8/9
140%
Type here to search
3:56 AM
4/6/2026


Beehive Health Classification System
Select Audio File
Show FFT
Show MFCCs
Show Spectral Features
Figure 2
MFCCs
MFCC Coefficients
Time
0:00
0:50
1:40
2:30
3:20
4:10
100
0
−100
−200
−300
−400
−500
Type here to search
3:56 AM
4/6/2026


Beehive Health Classification System
Select Audio File
Show FFT
Show MFCCs
Show Spectral Features
Figure 3
Spectral Features
Spectral Centroid
Spectral Bandwidth
Spectral Rolloff
Zero-Crossing Rate
Value
Frames
16000
14000
12000
10000
8000
6000
4000
2000
0
5000
10000
15000
20000
25000
Type here to search
3:57 AM
4/6/2026



VIII. CONCLUSION

This project presents a novel approach to monitoring beehive health using acoustic signals and machine learning. By analyzing the natural sounds produced within a hive, the system provides a non-invasive and efficient method for detecting colony conditions. The integration of MFCC-based feature extraction and a Random Forest classifier enables accurate classification of hive health.The developed system demonstrates that audio-based monitoring can significantly reduce the need for manual inspection, minimizing disturbance to bees and improving productivity. The inclusion of visualization tools further enhances the system by providing insights into the acoustic characteristics of beehives.The proposed solution is cost-effective, scalable, and user-friendly, making it suitable for both small-scale and large-scale beekeeping operations. Future work can focus on integrating IoT devices for real-time monitoring and exploring deep learning models for improved accuracy.Overall, this system contributes to the advancement of precision apiculture and supports sustainable agricultural practices.

REFERENCES

1. Potamitis, I., et al. “Automatic Bee Hive Monitoring Using Acoustic Signals.”
2. Ferrari, S., et al. “Monitoring of Bee Colonies Using Sound Analysis.”
3. Nolasco, I., et al. “Machine Learning in Acoustic Beehive Monitoring.”
4. Vasconcelos, M., et al. “Audio-Based Classification of Bee Activity.”
5. Bishop, C. M. *Pattern Recognition and Machine Learning*.
6. Breiman, L. “Random Forests.” Machine Learning Journal.
7. Librosa Documentation – Audio Processing in Python
8. Scikit-learn Documentation – Machine Learning Library

9. IEEE Papers on Smart Agriculture and IoT
10. FAO Reports on Pollination and Agriculture
11. Goodfellow, I., et al. *Deep Learning*
12. Zhang, Y., et al. "Acoustic Signal Processing for Environmental Monitoring"
13. Kaggle Datasets – Bee Sound Classification
14. Python Tkinter GUI Documentation
15. ResearchGate Publications on Precision Apiculture