



Machine Learning-Based Classification of Mosquito Wing Beats Using Mel Spectrogram Images and Ensemble Modeling

Bandi Vamsi^{1*}, Ali Al Bataineh², Bhanu Prakash Doppala³

¹ Department of Artificial Intelligence, Madanapalle Institute of Technology & Science, Andhra Pradesh 517325, India

² Artificial Intelligence Center, Norwich University, Northfield VT 05663, United States

³ Data Analytics, Generation Australia, Sydney 2000, Australia

Corresponding Author Email: aalbatai@norwich.edu

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ts.410437>

ABSTRACT

Received: 22 November 2023

Revised: 17 February 2024

Accepted: 29 March 2024

Available online: 31 August 2024

Keywords:

convolution neural network (CNN), ensemble modeling, Mel spectrogram, mosquito wing beats

These days, many dreadful diseases are caused by mosquitoes, along with other types of infections. Mosquitoes are also called silent feeders. Due to this ability, mosquitoes take advantage of increasing their capacity to spread diseases. Many life-threatening diseases such as malaria, dengue, Zika, yellow fever, and chikungunya are caused by these mosquitoes. These diseases are caused by viruses, parasites, and bacterial pathogens through various vectors like *Aedes aegypti* and *Culex*. Due to the rapid increase in cases worldwide, there is a necessity to deploy an intelligent machine-automated model to decrease the spread of infections. The method used in this study detects different types of mosquitoes responsible for spreading these diseases. The key to controlling the spread of infection is to detect the type of mosquito based on the beat of its wings. The sound recordings related to mosquito wing beats, collected from different sources, are used in this study. These recordings are divided based on the mosquito species through max pooling and convolution models. The entire work is framed under three segments: identifying the recorded sound audio file to get a Mel spectrogram image, extracting features using pooling and convolution methods, and identifying the mosquito type through an ensemble method using classifiers like Random Forest, Support Vector Machine (SVM), and Decision Tree. The frequency waves are used to transform the audio recordings into spectrograms in the preprocessing phase. The spectrogram filter is used to eliminate noise from the spectrogram images. Vector values are obtained using pooling and convolution methods. The values from the classifiers used in this work are then fed into the ensemble method to identify the mosquito type based on its wing beats. Based on the final results and observations, the SVM classifier achieved the highest accuracy, with 95.05% for the type *Aedes albopictus*, compared to the other classifiers.

1. INTRODUCTION

Nowadays, fever is a common disease spreading globally. The main root cause of fever spread is mosquitoes [1]. Various kinds of fevers are spreading and infecting people through different types of mosquitoes. According to a 2019 report by the World Health Organization (WHO), almost 400,000 deaths were caused by malarial fever alone [2]. Particularly in India, deaths due to dengue fever are 17,546 per 100,000 people. Fevers, when they turn into serious infections, may lead to different life-threatening complications such as chikungunya, dengue, Zika, and yellow fever [3].

The main aim is to detect mosquito wing beat sounds during flight in an acoustic time series. The key scientific modules that support this work provide the necessary data for the development of machine learning models to identify and classify occurrences in the dataset. The outcomes obtained from this audio detection are generated, examined, and applied to the audio information gathered in the field [4].

Traditional examination approaches for malaria, such as

recording mosquitoes landing on human surfaces, consume more time, space, and money. The research also addresses the complications of disease contraction, which are critical for elimination efforts [5]. Due to these challenges, many vector methods representing the geographical spread of these insects rely on sparse, unevenly divided bi-directional data [6].

However, the models used in this work are concerned with one such processing region, which faces typical risks such as data imbalance, inadequate information, low transmission rates, selection bias, and varying labels. Therefore, this study considers other contexts and broader detection conditions, such as acoustic time series information [7].

The aim of this work is to develop a proper mechanism to detect the type of mosquito based on the breed that spreads malaria, using their wing beat features [8]. By doing so, we can implement the necessary structure to aid malaria-infected areas where it is needed the most. Additionally, these models can enhance audio machine learning algorithms. These enhancements are crucial in the study of feature extraction, using deep learning methods to work with widely available yet

imperfectly labelled data [9, 10].

This work necessitates an essential automatic information system for detecting and classifying various mosquito species to aid in targeting the crucial areas affected by disease spread [11, 12]. With the help of these technologies, health organizations and other stakeholder groups can effectively identify disease-spreading mosquitoes and focus on infection hotspots [13, 14]. By understanding the dangerous threats and the increase in infections, the implemented techniques successfully help epidemiologists enhance scientific conclusions and methodologies [15]. Officials can be notified to take preventive measures in controlling the spread of infections by following specific steps in the infected regions [16, 17]. In this research, a hybrid machine learning algorithm is proposed for classifying mosquito species based on their wing beat sounds.

To detect the stream of audio, the algorithm, with the help of machine learning, is implemented in a three-step succession. The components are used to extract the audio signal data for a required time period. This data is then converted into necessary frequency components such as frequency and amplitude [18]. The spectral form is generally used as input to the network for noise identification. The type of spectroscopic format utilized significantly affects the efficiency of the network methods. Classifiers trained through this method represent human perception due to the Mel Frequency Cepstral Coefficient (MFCC), which gathers data at shorter wavelengths that are recognizable to humans [19, 20]. Consequently, the MFCC is widely used in this study as a primary spatial frequency factor. The workflow through machine learning to classify mosquito types based on the wing beat audio file is represented in Figure 1.

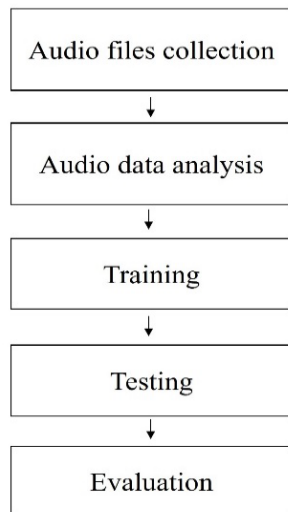


Figure 1. Work flow of ML for classification

In this research, we also considered the functioning of MFCC to feed data into the classifier. The extraction of features converts the MFCC data into a feature format for the audio signal to detect the signalling class. Obtaining the average vector values of spectral information is a standard phase in characteristic selection models.

The main objectives of this work are divided into following stages:

a) Firstly, the sound of a wingbeats is transformed into wave forms through black and white channels (binary) from the raw audio data.

b) The Mel spectrogram is used to remove the embedded noise which is required in maintaining the frequency formats of RGB image.

c) This modified image is now sent to CNN architecture for retrieving the embedded features with the help of convolutions.

d) Through fully connected layer, the final vector values are obtained and are given to ML classifiers for classifying the intensity of wingbeats of mosquitoes depending upon their range of frequencies.

The remaining sections of this is framed as: Section-2 discusses about ‘Related Works’, Section-3 discusses about ‘Methodology’, section-4 for describing ‘Results and its discussions and section-5 for ‘Conclusion with link to Future Scope’.

2. RELATED WORKS

The connection uses max-pooling to determine if a particular factor is present in the image. Once the image is loaded, the required location of data is erased, indicating that a signal has been detected. The exact placement of the detected factor is less likely to be determined relative to its initial origin concerning other factors. This is primarily because there are fewer cumulative attributes, which reduces the number of required features in the hidden region—an added advantage. For instance, in the case of pictures and different forms of information, pooling is used to make the method compatible with minor transformations of the given input [21].

The derived factors in the framework show an increase in the exceptional determination of information through the alternating use of convolution layers and pooling methods. Since each metric utilizes a large portion of the data, the layering arrangement is more effective when compared to larger instances. These methods are widely used in audio evaluation and various other technical studies, providing high accuracy for maintaining higher resolution division, as demonstrated by the results [22].

Considering the expertise in mosquito identification, computer vision relies on structural factors to assign a specific binomial nomenclature. Machine learning uses this phase as a systematic method for accurate insect identification. Park et al. demonstrated an outstanding presentation using convolutions, achieving a 97% accurate result from eight gathered samples, which included two specific species from different regions. Additionally, species that shared common features were divided and grouped together. By considering all varieties of species through annotations, the same species were accurately identified [23].

Using Convolutional Neural Networks (CNNs), were able to compare 17 classes with an efficient accuracy rate of 96.96%. The division of closely connected morphological features in the species is challenging to extract into their respective classes. The study made extensive use of colony observations to address the grouping of mosquito samples for procuring image-based information. The inbreeding headquarters often lose or terminate certain features that are not present in the initial samples. The data is particularly clean, especially when compared to the flawed measurements collected through suction complexities in the field of study [24].

The process of considering the outcomes from CNN input and classifiers is a method to identify the species. By taking

the result as a prior probability, the performance rate is 80.8%. The model with VGG-13 and a 2D-ConvFilter of dimension 3x3 is used to identify the type of mosquitoes. Additionally, an average improvement of 5.5% is observed when cyclical cycle information is incorporated into the network. With the addition of cyclical data, the VGG-13 architecture with a 1D-ConvFilter achieved a higher performance rate of 85.7% [25].

The CNN model, previously implemented on VGG-16 [26], to differentiate various types of mosquitoes. Aedes and other types of mosquitoes were used as samples in their study. A wide range of epoch times was utilized in the research. This model achieved a maximum accuracy of 97% based on the initial restricted features. Consequently, vector detection and efficient disinfection were effectively implemented in the study.

A section of work [27] involved the visual recording of six species of mosquitoes. This study evaluated the characteristic features and identified the sound of the insects based on their wing beats. The research considered 279,566 acoustic recordings of flying mosquitoes. The model demonstrated an accuracy of 96% using advanced deep learning techniques.

Using CNN, Schreiber et al. divided the vector values of mosquito species based on the sound of their wing beats recorded with electrical devices. With the help of a spectrum analyzer, the velocity of the wing beat sound was estimated based on category, multiclass, and a certain group of classifiers. The efficiency of the classifiers used in the study was examined under extremely noisy conditions. The noise could be reduced since the audio recordings were arranged in a setting with minimal background noise. The method used in the study is very expensive, as the sensors used in groups are difficult to manage, and obtaining high-quality audio recordings is a challenging task [28].

Images of mosquitoes, the researches [29, 30] developed larva-based CNN classification models. With a limited dataset and after careful tuning, they observed good classification performance with previously trained CNNs. This technique is used to attain a vector from the sound of mosquito wing beats. Based on these methodologies, healthcare providers and taxonomists have not yet evaluated the study due to the collected information and research performed during experimentation.

The wing beat sound from the remaining audio clip. The sound waveform was converted into a spectrum using Continuous Wavelet Transformation. The amplitude spectrum of the wing beat sound was processed using a Gaussian function, and then the weights and splits were evaluated. Additionally, the frequency of the sound was analysed through a multivariate sampling distribution developed from the spectrum. To evaluate the probability of various mosquito audio flying sounds, different methods were utilized. The research, which analysed the wing beat sounds of six mosquito species, achieved an accuracy of 95.23% [31].

The existing studies [21-31] demonstrate that automatic classification of mosquito types can be achieved through the audio sound of their wing beats and sensitivity models. The methodologies used involve elevating the wavelength of wing beats through pseudo acoustic sensor devices. Acoustic-based models, such as wavelet transform or spectrum analysers, are used to extract audio files or produce noise-removed modified signals of the wing beats. This process contrasts with using mosquito images.

Each audio file has a duration of 0.01 to 0.03 seconds, which is very difficult for the human ear to detect the mosquito

species. Considering this limitation, a preprocessing methodology is used in this study to process such short-duration audio file formats. The audio files are converted into waveforms to obtain frequencies. These frequencies are then transformed into Mel spectrograms, converting the audio files into image formats.

3. METHODOLOGY

3.1 Dataset

In this study, data is used from the Kaggle repository [32]. This dataset contains information related to different varieties of mosquitoes along with their wing beat sounds. It includes 279,566 recordings of wing beats pertaining to six mosquito species: *Aedes aegypti*, *Aedes albopictus*, *Anopheles arabiensis*, *Anopheles gambiae*, *Culex pipiens*, and *Culex*.

The proposed model used in this work employs a group of informational techniques to ensure that the evaluated results meet a high standard of accuracy. Additionally, it is designed to identify different methods to process audio format data with minimal computational effort. The feature extraction model developed in this study plays a crucial role in classification, selecting the required and necessary traits. A sample summary of the information used in this research work is shown in Table 1.

Table 1. Summary of mosquito wing beat recordings [31]

S. No	Mosquito Type	File Size	Audio Duration with Noise	Audio Duration After Preprocessing
1	Aegypti	9.8 KB	0:05 sec	0:02 sec
2	Albopictus	9.8 KB	0:10 sec	0:02 sec
3	Arabiensis	9.8 KB	0:12 sec	0:02 sec
4	Gambiae	9.8 KB	0:04 sec	0:02 sec
5	Pipiens	9.8 KB	0:06 sec	0:02 sec
6	Culex	9.8 KB	0:11 sec	0:02 sec

3.2 Analysis of audio data

Audio signals are represented through sound waves, which can propagate through various transmission media such as gas, liquid, or as ultrasonic waves. In terms of magnitude and time, these signals typically remain constant. They can be quantized in two dimensions to be arranged digitally as time-series data.

A time series is defined as a sequence of data points labeled by time, making it a group of random values. The theoretical approach involves a constant alignment of information with exponential factors such as mean and variance, which may vary depending on the time interval. The samples are quantized to their nearest values within the group of digital ranges. This process of digital transmission speed is commonly grouped as periodic and associated. The accurate conversion of spectrum signals at a certain sampling frequency is twice the resonance value of relevance, referred to as sampling. During the encoding process, the calculation of amplitude generates intermittent values.

The final outcomes of raw frequencies are extracted and transformed into a waveform in WAV format, as shown in Figure 2. Ranging from 440 Hz to 5.1 KHz, the sound wave in WAV format of a mosquito over 5 seconds is represented in Figure 3. The image factors in the audio and every change in the waveform are explained in the sections below with the help of the mentioned figures.

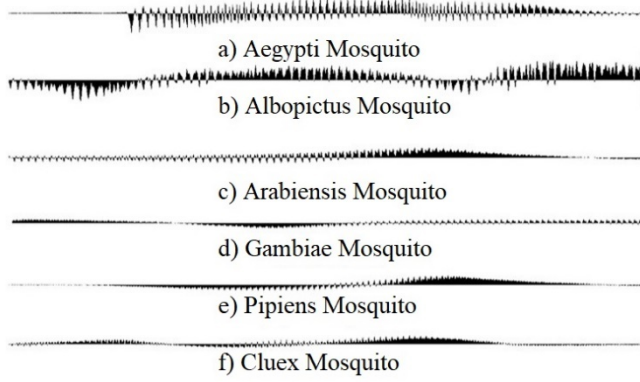


Figure 2. Wave form in WAV of mosquito wing beats

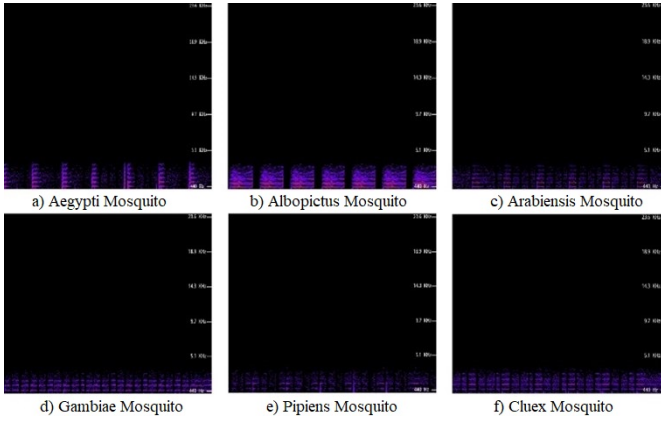


Figure 3. Mel spectrograms of mosquito wing beats

Intermediate methods are commonly deployed through electronic sound waves. These methods are utilized as data in the evaluation of acoustic factors. Generally, these representations accurately record how humans recognize audio signals by altering the division of frequencies over time. They rely on both low- and high-level acoustic estimations, as well as musical presentations, and thus mid-level features have their specific term.

The intermediate concepts broadly used in the study are frequency and time. These constitute the amplitude and energy required for transmission at different frequencies. Frequencies vary over time along a wider timeline, and they perform transmission as the magnitude also changes. These predictions, in reality, act as a substitute for frequency in terms of accuracy with time resolution.

The commonly used time-frequency forms are developed using a spectrum analyser. The execution of a Short Time Wavelet Transformation (STWT) with initial frames attempts to stabilize and generate a waveform using Fourier Transform (FT). The energy in various bandwidths among the transformed frames is initiated through electric spectra, which are combined to develop a single frame. The derived characteristics are produced by converting the resulting image into a vector form. Generally, a Fourier Transform is considered the exceptional transformation from which most intermediate forms are derived. This is shown in Eq. (1) to brief the STFT along with some perceptions.

$$FT\{x(t)\} = \int_{-\infty}^{+\infty} x(t) e^{-i\omega t} dt \quad (1)$$

The functioning of a frame that is positive for a particular period of time consists of a variable that needs to be transformed, $x(t)$. Considering the Fourier Transform of the complete signal under a single frame, it is progressed with a time vector to produce a two-dimensional representation of the data. For enhancing the characteristic features during the post-processing phase, the spectrograms are significantly modified. To show the corresponding volumes at different frequencies, the wavelengths present in the spectrum analyser are converted to a logarithmic scale.

Generally, a randomly selected variable is chosen for time-series data. This is done among samples that are close in time, making them more likely to be connected, compared to those that are farther apart. Additionally, time-series methods are utilized in chronological pairing, where the allotted time value has some connection with previous data, rather than with isolated values.

A signal is defined as data that represent compulsive phenomena. The audio channel utilized in the study is represented as a sample of metadata. The main aim of the study is to extract signals from audio data, which typically contain a significant amount of noise. Since noise is present in almost all real-world fields, it is important for machine learning to have the capability to recognize and address this concept.

Noise is referred to as any unreliable or unconnected data present in the evaluation. Noise is typically produced due to obstructions during the recording process. In simple terms, noise is a factor that introduces irregularity in the digital signal. It does not contain any information regarding the crucial variables used in the study. The correlation function value $R_s(T)$ is the power of a signal $s(t)$ when it in imaginary process that is constant is defined in Eq. (2):

$$R_s(T) = \text{Expected Value } (s(t) * s(t + T)) \quad (2)$$

Related connections are managed between its correlation function and the power of noise P_n is represented in Eq. (3):

$$P_n = \text{Expected Value } (n^2(t)) \quad (3)$$

3.3 Proposed model

Our proposed model is illustrated in the workflow by organizing different kinds of mosquitoes based on the sound of their wing beats, as shown in Figure 4. The methodology is divided into three sections: Firstly, the pre-processing phase where the raw audio is processed; and finally, with the help of classifiers like RF (Random Forest), SVM (Support Vector Machine), DF (Decision Forest), and CNN (Convolutional Neural Network), the spectrogram images are processed.

During the pre-processing phase, the raw audio data is given as input to identify the intensity of the mosquito wing beat audio. All the intensities obtained from different audio files are grouped to develop a waveform in WAV format. There may be a chance of missing frequencies due to background noise. To eliminate this noise, Mel spectrogram filters are utilized to maintain the quality of the audio data. After this, the smooth waveform obtained is converted to spectrograms by producing a re-sampled frequency with a range of 440 Hz to 5.1 KHz.

In the second phase, the raw image format file is given as input to the CNN technique for feature extraction. Convolutions are developed during the encoding process. They are utilized in the CNN methodology to reduce the spectrogram image size. To minimize the number of shift

values in a hidden layer, a batch normalization layer is added for each pooling and convolutional pair. This helps to increase the learning speed and reduce overfitting. Once the pooling process is completed, a layer called the flatten layer is produced. To further reduce overfitting, the flatten layer maintains a dropout rate of 50%. The final layer, called the fully connected layer, integrates an activation process, making it robust.

In the final phase, the established features from the CNN architecture are given to classifiers as input. The classifiers SVM, DT, and RF are used in the methodology to classify the sound of the mosquito. An ensemble model is used in this phase along with the classifiers. Based on the outcome of each classifier, the accuracy is detected and the recorded sound of the mosquito wing beat is identified.

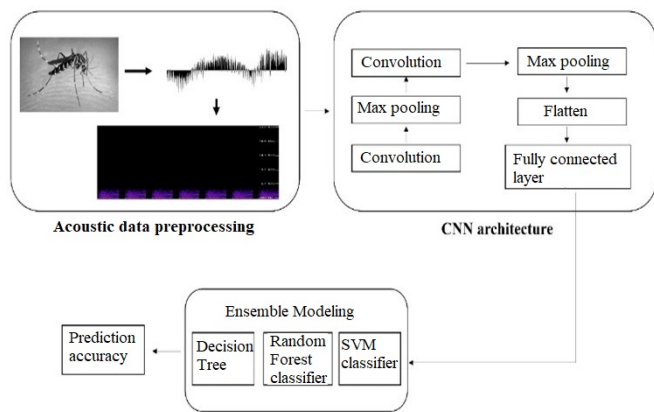


Figure 4. Proposed methodology

3.4 Decision tree (DT)

In a decision tree (DT), the non-leaf nodes represent quality checks on the applied factors, each branch shows the outcome of the given input, and every terminal node includes a classifier. The main root of the decision tree is the node located at the top of the tree structure. The decision tree is formed by dividing the dataset, which acts as the structure of the tree. The primary reason for this division is the collection of various feature nodes based on classification features. For every developed subset in the tree, recursive partitioning is performed. The Gini Index can be calculated, by considering the probability P_i of the chosen node having the label i along with the classification error for that node ' $\sum_{k \neq i} P_k$ '. This becomes zero when every sample in the node support within a single region. Let us suppose P_i is proportion of the nodes in a given group of k items in the same class with i label which is given by Eq. (4):

$$\text{Gini Index} = \sum_{i=1}^k P_i \sum_{K \neq i} P_k \quad (4)$$

A disadvantage of conventional methods in implementing decision trees is that when the branches of the tree have deep roots, they may extend to overfitting the corresponding training data, resulting in limited bias but increased variance. To diminish this variation, the Random Forest algorithm groups multiple hierarchical decision trees constructed on different portions of the same dataset. This technique helps enhance the accuracy of our developed model. However, the

total cost of the model sees a slight rise in bias and also experiences a slight decrease in accuracy.

3.5 Random Forest (RF)

The commonly used model of resampling, also known as bootstrapping, is employed in the Random Forest model for identification. This prediction method in the classifier is used to build the decision trees. Bootstrapping constantly among the randomly selected by restoring the test phase through a training dataset set $X = \{X_1, X_2, X_3, \dots, X_n\}$ with solutions $Y = \{Y_1, Y_2, Y_3, \dots, Y_n\}$ develop a tree for all the selection. The training variables ' X_b ' and ' Y_b ' are relied on a classification tree. Gathering the highest count of votes from every individual classification method on sampling after training, identification for unknown data X can be evolved.

The bootstrapping model decreases the variance without affecting the bias. If the bias remains unchanged, the model works efficiently. Consequently, single tree detection is accurate in the training dataset. The mean of various other trees is calculated when the trees are irrelevant. By using different training datasets for trees, the random sampling model interlinks the nodes, which results from training a number of trees in a single phase of training.

3.6 Support vector machine (SVM)

The SVM, also known as a quadratic classifier, shows an ideal representation by having a higher dimensional space among the subclasses. The basic functions represent the maximum distance between the regions and the hyperplane. Tolerance is allowed for a few misclassifications that are out of range, evaluated under the border range of the SVM classifier. The SVM classifier is related to kernel models that encompass a wide class of methodologies, which use kernel methods to transform features into a high-dimensional training dataset. Consequently, a hyperplane is plotted along the entire process of the training dataset. The classifier has unpredictable division rules. Considering the binary classification problem under the construction of linear methods is represented in Eq. (5).

$$y(x) = W^T \phi(X) + b \quad (5)$$

We have the final biased parameter ' b ' and represented ' $\phi(X)$ ' as a translation of feature extractor. Keen observation has to be done so that the dual method eliminates working in feature set directly and is outlined in terms of basic functions.

The input vectors $x_1, x_2, x_3, \dots, x_n$ with matching detected values $t_1, t_2, t_3, \dots, t_n$ build the initial sample, and new data points x are detected relying on the indication of $y(x)$. Since the trained model is differentiable in higher dimensional space, there must have at least one combination of the specifications w and b that allows a component of Eq. (6) to satisfy $y(x_n) > 0$ for observations with $t_n = +1$ and $y(x_n) < 0$ for points with $t_n = -1$, resulting in $t_n y(x_n) > 0$ for all training samples.

The border range is calculated as the shortest distance between the decision function and each observation. This is how the SVM addresses this problem. In SVMs, the margin-maximized decision function is chosen as the classifier, as given by Eq. (6).

$$t_n(W^T \phi(X_n) + b) \geq 1, \text{ where } n = 1, \dots, N \quad (6)$$

3.7 Convolutional neural network (CNN)

Every phase in a feed-forward deep network comprises a set of neurons. These neurons evaluate a non-linear element. This evaluation of the input layer is done after finding the linear combination of weights, $z = W^T X + b$. Thus, the parameters of the deep network method, using weights and biases, are developed. The group of sigmoid functions that contain the logistic function is a common choice for the activation function, as represented in Eq. (7).

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (7)$$

A fully connected network with at least one convolutional layer is defined as a Convolutional Neural Network (CNN). CNNs use convolutions instead of ordinary complex numbers during the process. They are primarily used to represent hierarchical methods, such as the grid organization of pixel intensities or commonly processed streams of audio. CNNs are useful in combining neighbouring spatial data in every medium. They contain convolution and pooling layers with fixed interconnections. The process of convolution generates a stack of extracted characteristic mappings. The input layers of the characteristic space are grouped together using convolution filters. To achieve efficiency, the number is substituted in the group of these convolutional layers using a complex number $W^l a^{(l-1)}$. The output of the signal, having at least one feature map a^l is shown mathematically in Eq. (8) as a set, where the convolution filters are denoted by W and the feature map is denoted by $a^{(l-1)}$.

$$a^l = \sigma(b + W^l a^{(l-1)}) \quad (8)$$

The execution of a neural network through convolution is represented in Figure 5. It is parameterized to detect two classes from a three-channel image (RGB) with a size of 32×32 . The outcome of the convolution having filtering process denoted by W is shown in Eq. (9) for every j and k^{th} neurons located in the hidden layer of the classifier.

$$\sigma(b + \sum_{l=1}^n \sum_{m=1}^n w_{l,m} a_j + l, k + m) \quad (9)$$

where, $a_{x,y}$ represents the input activator at certain point, $w_{l,m}$ is a $n \times n$ array of fully convolutional, and b is the core value for the bias (x, y) .

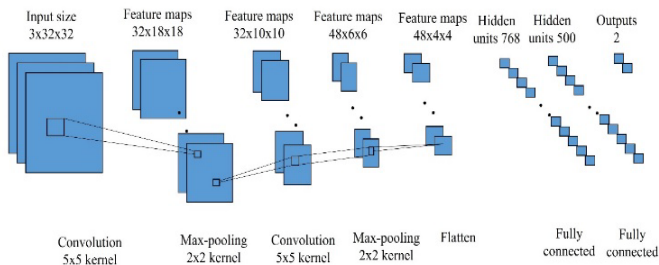


Figure 5. An RGB of three-channel with 32×32 image

To gain an advantage through the input configuration and reduce the number of attributes, the connection gradient is

modified by changing the product vector values in the fully connected convolution layer. Each module is duplicated across the entire input, with a specific sample of the module used in the back portions of the layer. The learning capability of the proposed model to handle greater input dimensionality is significantly improved by reducing the features.

The convolutions regularly include pooling layers interspersed among them to reduce the complexity of the derived features. Since the fully connected layer is only evaluated through local connections given as input, an increase in the layers inversely affects the precision. This shows a connection among a larger section of the input. To reduce the complexity of a classifier, a max pooling technique achieves data gain over random local input regions.

The feature parameters, filter sizes and drop out layer details are given in Table 2. The total parameters of our proposed model are 3,17,706 and all these parameters found trainable.

Table 2. Sequential model summary

Layer	Output Shape	Parameters
Conv2D	(26,26,64)	640
Conv2D	(26,26,64)	36928
Max_pooling_1	(12,12,64)	0
Conv2D	(10,10,128)	73856
Conv2D	(8,8,64)	73792
Max_pooling_2	(4,4,64)	0
Dropout	(4,4,64)	0
Flatten	1024	0
Dense	128	131200
Dense_1	10	1290

4. RESULTS AND DISCUSSIONS

4.1 Classification metrics in machine learning

4.1.1 Accuracy

The easiest and most effective metric for classification is accuracy. This is calculated as the ratio of the number of accurate predictions to the total number of predictions, as shown in Eq. (10).

$$Accuracy = \frac{\text{Number of corrected predictions}}{\text{Total number of predictions}} \quad (10)$$

4.1.2 Precision

Precision is defined as the fraction of positive predictions that are accurate. It is calculated as the ratio of true positives to the sum of true positives and false positives, as determined in Eq. (11).

$$Precision = \frac{\text{True positives}}{\text{True positives} + \text{False positives}} \quad (11)$$

4.1.3 Sensitivity

It represents the percentage of actual positive samples that are incorrectly identified. True Positives are calculated as the total number of positive samples that are correctly identified as positive or inaccurately identified as negative. These are factually correct in relation to the total number of predictions identified as positive, as given by Eq. (12).

$$Sensitivity = \frac{\text{True positives}}{\text{True positives} + \text{False negatives}} \quad (12)$$

4.1.4 Specificity

Specificity refers to the score that measures the model's ability to correctly identify true negatives for each classification. It is calculated using Eq. (13).

$$\text{Specificity} = \frac{\text{True negatives}}{\text{True negatives} + \text{False positives}} \quad (13)$$

4.2 Performance of proposed model

The implemented model uses a distinct approach to identify classified outcomes. The classification results for identifying mosquitoes through wing beat sound using CNN are described in Table 3. Tables 4, 5, and 6 present the results of mosquito identification through wing beat sound using CNN with the classifiers RF, SVM, and DT, respectively.

Table 3. Classification outcome of CNN

Model	Mosquito Type	Accuracy	Precision	Sensitivity	Specificity
CNN	Aegypti	92.19	90.29	95.12	45.21
	Albopictus	91.05	89.15	93.95	45.71
	Arabiensis	91.32	89.46	94.13	46.76
	Gambiae	89.17	89.16	93.74	44.42
	Pipiens	90.53	88.66	93.49	44.39
	Culex	90.69	88.88	93.71	45.21

Table 4. Classification outcome of CNN+DT

Model	Mosquito Type	Accuracy	Precision	Sensitivity	Specificity
CNN+DT	Aegypti	91.69	90.1	94.59	63.21
	Albopictus	92.19	90.6	95.06	63.71
	Arabiensis	91.46	89.91	94.24	64.76
	Gambiae	89.01	89.31	93.55	62.42
	Pipiens	90.84	89.28	93.77	62.39
	Culex	91.6	90.1	94.59	63.21

Table 5. Classification outcome of CNN+RF

Model	Mosquito Type	Accuracy	Precision	Sensitivity	Specificity
CNN+RF	Aegypti	94.46	92.52	97.35	44.61
	Albopictus	94.96	93.02	97.82	45.11
	Arabiensis	94.23	92.33	97.10	46.16
	Gambiae	91.78	91.73	96.31	43.82
	Pipiens	93.61	91.7	96.53	43.79
	Culex	94.37	92.52	97.35	44.61

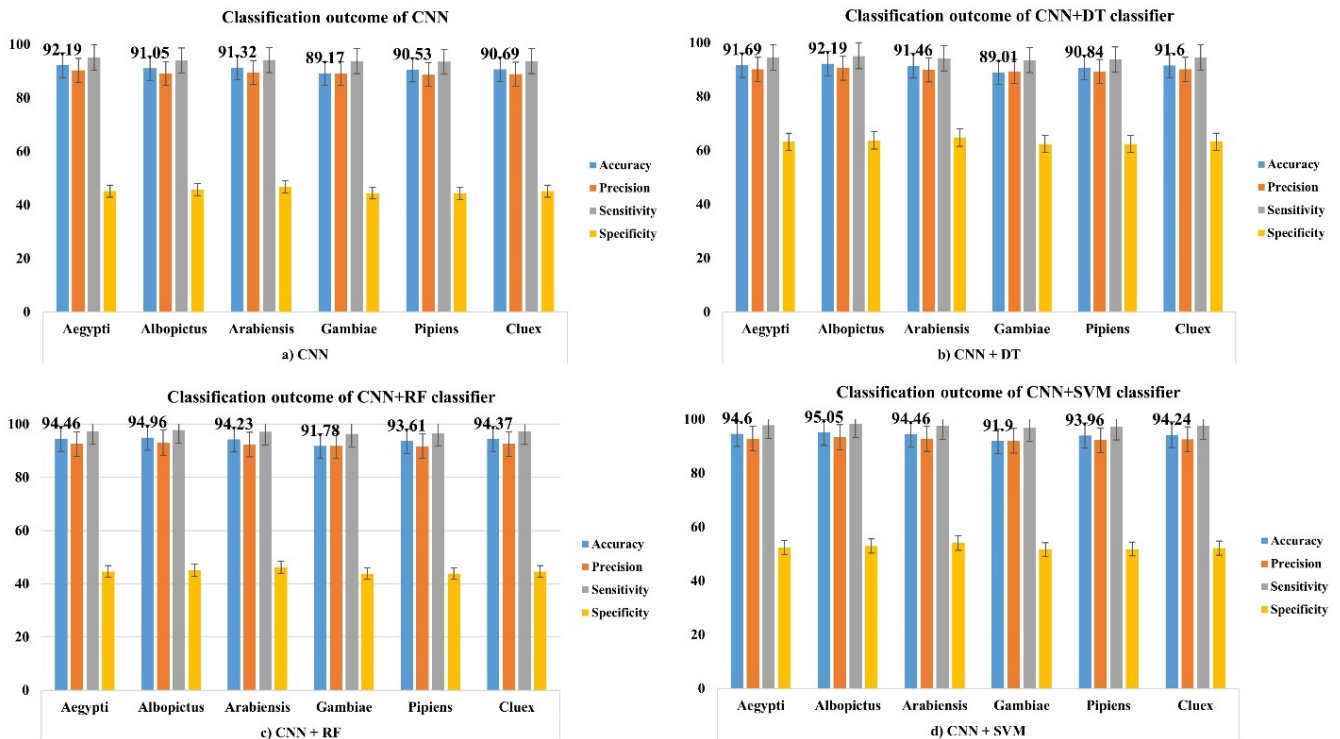


Figure 6. Graphical representation of classification outcomes

Table 6. Classification outcome of CNN+SVM

Model	Mosquito Type	Accuracy	Precision	Sensitivity	Specificity
CNN+SVM	Aegypti	94.6	92.84	97.77	52.42
	Albopictus	95.05	93.29	98.19	52.87
	Arabiensis	94.46	92.74	97.51	54.06
	Gambiae	91.9	92.03	96.71	51.61
	Pipiens	93.96	92.23	97.16	51.81
	Culex	94.24	92.57	97.5	52.15

In this work, six different types of mosquito wing beat audio recordings are considered for classification to achieve quality outcomes beneficial for the research. These classification results are presented visually and shown in Figure 6.

From this analysis, it can be concluded that the CNN architecture requires substantial GPU processing power to extract features embedded in the spectrograms. Consequently, classification with this architecture takes more time to identify mosquitoes based on wing beat sounds. Based on this observation, classification in the fully connected layers can be performed using machine learning classifiers. This approach overcomes the limitations of heavy models and introduces greater classification accuracy.

5. CONCLUSION AND FUTURE SCOPE

This work aims to identify mosquito types using a CNN combined with machine learning classifier methods. Initially, the data obtained from the preprocessing phase is converted into waveform format. These waveforms are then transformed into spectrogram images using normalized frequency. The spectrogram images are fed into the CNN architecture, which extracts the required features through pooling and convolution operations. These features are then passed to various classifiers used in the study to detect mosquito types based on their wing beats.

Observations show that SVM achieved higher accuracy, whereas RF showed lower negative predictions. The sound recordings used have a short duration, ranging from 0:01 to 0:03 seconds. To handle such short-duration audio files, memory-based frameworks are required to detect and identify different mosquito species globally. The basic CNN architecture demonstrated accurate results with the limited classified data available. To include mosquito types from various regions around the world, memory-based methods need to be developed. Future work will involve implementing a storage-based automated memory method to save vector data of different mosquito wing beats under all required conditions.

REFERENCES

- [1] Montgomery, D.C., Jennings, C.L., Kulahci, M. (2015). Introduction to Time Series Analysis and Forecasting. John Wiley & Sons.
- [2] Gemmeke, J.F., Ellis, D.P., Freedman, D., Jansen, A., Lawrence, W., Moore, R.C., Plakal, M., Ritter, M. (2017). Audio set: An ontology and human-labeled dataset for audio events. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Orleans, LA, USA, pp. 776-780. <https://doi.org/10.1109/ICASSP.2017.7952261>
- [3] Pons, J., Serra, X. (2019). Randomly weighted cnns for (music) audio classification. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, pp. 336-340. <https://doi.org/10.1109/ICASSP.2019.8682912>
- [4] Koutini, K., Eghbal-Zadeh, H., Dorfer, M., Widmer, G. (2019). The receptive field as a regularizer in deep convolutional neural networks for acoustic scene classification. In 2019 27th European Signal Processing Conference (EUSIPCO), Coruna, Spain, pp. 1-5. <https://doi.org/10.23919/EUSIPCO.2019.8902732>
- [5] Lee, J., Kim, T., Park, J., Nam, J. (2017). Raw waveform-based audio classification using sample-level CNN architectures. arXiv preprint arXiv: 1712.00866. <https://doi.org/10.48550/arXiv.1712.00866>
- [6] Kim, T., Lee, J., Nam, J. (2019). Comparison and analysis of SampleCNN architectures for audio classification. IEEE Journal of Selected Topics in Signal Processing, 13(2): 285-297. <https://doi.org/10.1109/JSTSP.2019.2909479>
- [7] Jeong, I.Y., Lee, S., Han, Y., Lee, K. (2017). Audio event detection using multiple-input convolutional neural network. In Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE), Munich, Germany, pp. 51-54.
- [8] Motta, D., Santos, A.Á.B., Machado, B.A.S., Ribeiro-Filho, O.G.V., Camargo, L.O.A., Valdenegro-Toro, M.A., Kirchner, F., Badaró, R. (2020). Optimization of convolutional neural network hyperparameters for automatic classification of adult mosquitoes. Plos One, 15(7): e0234959. <https://doi.org/10.1371/journal.pone.0234959>
- [9] Pataki, B.A., Garriga, J., Eritja, R., Palmer, J.R., Bartumeus, F., Csabai, I. (2021). Deep learning identification for citizen science surveillance of tiger mosquitoes. Scientific Reports, 11(1): 4718. <https://doi.org/10.1038/s41598-021-83657-4>
- [10] Vijayalakshmi, A. (2020). Deep learning approach to detect malaria from microscopic images. Multimedia Tools and Applications, 79(21): 15297-15317. <https://doi.org/10.1007/s11042-019-7162-y>
- [11] Park, J., Kim, D.I., Choi, B., Kang, W., Kwon, H.W. (2020). Classification and morphological analysis of vector mosquitoes using deep convolutional neural networks. Scientific Reports, 10(1): 1-12. <https://doi.org/10.1038/s41598-020-57875-1>
- [12] González-Pérez, M.I., Faulhaber, B., Williams, M., Brosa, J., Aranda, C., Pujol, N., Verdún, M., Villalonga, P., Encarnação, J., Busquets, N., Talavera, S. (2022). A novel optical sensor system for the automatic classification of mosquitoes by genus and sex with high levels of accuracy. Parasites & Vectors, 15(1): 190. <https://doi.org/10.1186/s13071-022-05324-5>
- [13] Akter, M., Hossain, M.S., Ahmed, T.U., Andersson, K. (2020). Mosquito classification using convolutional neural network with data augmentation. In International Conference on Intelligent Computing & Optimization.

- Springer, Cham, pp. 865-879. https://doi.org/10.1007/978-3-030-68154-8_74
- [14] De Los Reyes, A.M.M., Reyes, A.C.A., Torres, J.L., Padilla, D.A., Villaverde, J. (2016). Detection of *Aedes Aegypti* mosquito by digital image processing techniques and support vector machine. In 2016 IEEE Region 10 Conference (TENCON), Singapore, pp. 2342-2345. <https://doi.org/10.1109/TENCON.2016.7848448>
- [15] Arista-Jalife, A., Nakano, M., Garcia-Nonoal, Z., Robles-Camarillo, D., Perez-Meana, H., Arista-Viveros, H.A. (2020). *Aedes* mosquito detection in its larval stage using deep neural networks. *Knowledge-Based Systems*, 189: 104841. <https://doi.org/10.1016/j.knosys.2019.07.012>
- [16] Mulchandani, P., Siddiqui, M.U., Kanani, P. (2019). Real-time mosquito species identification using deep learning techniques. *International Journal of Engineering and Advanced Technology*, 2249-8958. <https://doi.org/10.35940/ijeat.B2929.129219>
- [17] Vasconcelos, D., Nunes, N., Ribeiro, M., Prandi, C., Rogers, A. (2019). Locomobis: A low-cost acoustic-based sensing system to monitor and classify mosquitoes. In 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), Vegas, NV, USA, pp. 1-6. <https://doi.org/10.1109/CCNC.2019.8651767>
- [18] Khalighifar, A., Jiménez-García, D., Campbell, L.P., Ahadji-Dabla, K.M., Aboagye-Antwi, F., Ibarra-Juárez, L.A., Peterson, A.T. (2022). Application of deep learning to community-science-based mosquito monitoring and detection of novel species. *Journal of Medical Entomology*, 59(1): 355-362. <https://doi.org/10.1093/jme/tjab161>
- [19] Yin, M.S., Haddawy, P., Nirandmongkol, B., Kongthaworn, T., Chaisumritchoke, C., Supratak, A., Sangmuang, C., Sriwichai, P. (2021). A lightweight deep learning approach to mosquito classification from wingbeat sounds. In *Proceedings of the Conference on Information Technology for Social Good*, pp. 37-42. <https://doi.org/10.1145/3462203.3475908>
- [20] Garcia, P.S.C., Martins, R., Coelho, G.L.L.M., Cámara-Chávez, G. (2019). Acquisition of digital images and identification of *Aedes Aegypti* mosquito eggs using classification and deep learning. In 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Rio de Janeiro, Brazil, pp. 47-53. <https://doi.org/10.1109/SIBGRAPI.2019.00015>
- [21] Box, G.E., Jenkins, G.M., Reinsel, G.C., Ljung, G.M. (2015). *Time Series Analysis: Forecasting and Control*. John Wiley & Sons.
- [22] Hershey, S., Chaudhuri, S., Ellis, D.P., Gemmeke, J.F., Jansen, A., Moore, R.C., Plakal, M.J., Platt, D., Saurous, R.A., Seybold, B., Slaney, M., Weiss, R.J., Wilson, K. (2017). CNN architectures for large-scale audio classification. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, USA, pp. 131-135. <https://doi.org/10.1109/ICASSP.2017.7952132>
- [23] Couret, J., Moreira, D.C., Bernier, D., Loberti, A.M., Dotson, E.M., Alvarez, M. (2020). Delimiting cryptic morphological variation among human malaria vector species using convolutional neural networks. *PLoS Neglected Tropical Diseases*, 14(12): e0008904. <https://doi.org/10.1371/journal.pntd.0008904>
- [24] Rustam, F., Reshi, A.A., Aljedaani, W., Alhossan, A., Ishaq, A., Shafi, S., Lee, E., Alrabiah, Z., Alsuailem, H., Ahmad, A., Rupapara, V. (2022). Vector mosquito image classification using novel RIFS feature selection and machine learning models for disease epidemiology. *Saudi Journal of Biological Sciences*, 29(1): 583-594. <https://doi.org/10.1016/j.sjbs.2021.09.021>
- [25] Ortiz, A.S., Miyatake, M.N., Tünnermann, H., Teramoto, T., Shouno, H. (2018). Mosquito larva classification based on a convolution neural network. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), pp. 320-325.
- [26] Fanioudakis, E., Geismar, M., Potamitis, I. (2018). Mosquito wingbeat analysis and classification using deep learning. In 2018 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, pp. 2410-2414. <https://doi.org/10.23919/EUSIPCO.2018.8553542>
- [27] Fernandes, M.S., Cordeiro, W., Recamonde-Mendoza, M. (2021). Detecting *Aedes aegypti* mosquitoes through audio classification with convolutional neural networks. *Computers in Biology and Medicine*, 129: 104152. <https://doi.org/10.1016/j.compbiomed.2020.104152>
- [28] Fuad, M.A.M., Ab Ghani, M.R., Ghazali, R., Izzuddin, T.A., Sulaima, M.F., Jano, Z., Sutikno, T. (2018). Training of convolutional neural network using transfer learning for *Aedes Aegypti* larvae. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 16(4): 1894-1900. <http://doi.org/10.12928/telkomnika.v16i4.8744>
- [29] Sanchez-Ortiz, A., Fierro-Radilla, A., Arista-Jalife, A., Cedillo-Hernandez, M., Nakano-Miyatake, M., Robles-Camarillo, D., Cuatrecasas-Jiménez, V. (2017). Mosquito larva classification method based on convolutional neural networks. In 2017 International Conference on Electronics, Communications and Computers (CONIELECOMP), Cholula, Mexico, pp. 1-6. <https://doi.org/10.1109/CONIELECOMP.2017.7891835>
- [30] Chen, Y., Why, A., Batista, G., Mafra-Neto, A., Keogh, E. (2014). Flying insect classification with inexpensive sensors. *Journal of Insect Behavior*, 27(5): 657-677. <https://doi.org/10.1007/s10905-014-9454-4>
- [31] Potamitis I., Rigakis I. (2016). Large aperture optoelectronic devices to record and time-stamp insects wingbeats. *IEEE Sensors Journal*, 16(15): 6053-6061. <http://doi.org/10.1109/JSEN.2016.2574762>
- [32] WINGBEATS. <https://www.kaggle.com/datasets/potamitis/wingbeats?resource=download>, accessed on 1 August 2024.