Mosquito Species and Larvae Classification Using a Modified Deep Learning Framework with XAI

Tahmid Enam Shrestha[1] [0009−0007−0075−5699] and M. Shamim Kaiser[2] [0000−0003−4721−8798]

Department of Computer Science, American International University-Bangladesh (AIUB), Dhaka, Bangladesh[1,2]
tahmidenam12@gmail.com[1] mskaiser@aiub.edu[2]

Abstract. Mosquito-borne infections, including malaria, dengue, and Zika virus, are substantial public health challenges worldwide. Early detection of mosquito species and larvae is essential for effective disease control and prevention. However, manual classification is labor-intensive and prone to error. Even though deep learning has advanced, existing automated classification models have suffered from poor performance, limited testing across diverse databases, and limited interpretability at high computational cost. This study presents a modified explainable deep learning model with an XAI approach to enhance transparency in decision-making. The model is trained on two datasets: Dataset A, containing images of mosquito species (Culex, Anopheles, Aedes), and Dataset B, focused on mosquito larval images. Data augmentation, such as resizing, rotation, and normalization techniques, is used to improve model generalization. The proposed integrated with XAI methods to achieve high classification accuracy and transparency. The effectiveness of the model is evaluated by standard metrics of accuracy, precision, recall and F1-score as compared to other existing state-of-the-art methods. The proposed model achieved test accuracies of 99.56% on Dataset A and 82.95% on Dataset B. The proposed model seeks to increase the sensitivity and specificity of mosquito classification by providing transparent, interpretable insights to support better public health surveillance and to implement more efficient mosquito control measures.

Keywords: Mosquito species classification · Larvae identification · Deep-learning · Explainable AI · CNN · Grad-CAM++ · Public health.

1 Introduction

Mosquito-borne diseases such as Malaria, Dengue, and Zika virus are a heavy burden on public health worldwide [25]. The timely recognition of the type and the invasive larval species is necessary for effective disease control. So far, identification of mosquito genera and larvae has been performed by visual observation, which is labor-intensive and error-prone.

However, recent advances in deep learning suggest the potential to automate this process by processing images of mosquito species and larvae [9, 21, 26, 14]. In spite of this, there are still many challenges in accurately categorizing mosquito species and their larvae, and in making this process interpretable and transparent. The objective of this study is to explore an explainable deep learning model with superior classification performance for mosquito species and larvae, incorporate Explainable AI (XAI) methods to enhance its interpretability, and evaluate its generalization ability across multiple datasets.

With the rapid development of deep learning, many automated solutions have been reported for mosquito species and larva recognition. The most commonly employed model is the CNN, which has demonstrated its powerful ability to learn high-level image features [20],[2]. Nonetheless, even with their success, substantial issues remain in mosquito systematics. Most current models perform well, but have shortcomings in cases of image quality variation and across species. Moreover, most works validate their models on a single data set, and generalization ability and stability cannot be guaranteed. Another issue is that most current models are non-interpretable, so one cannot understand why they make specific classification decisions [15],[10]. Lastly, hyperparameter tuning and model optimization are computationally intensive, which impedes the practical application of these models.

Therefore, a stronger and interpretable way of characterizing mosquito species and larvae is required, one that can be generalized easily, which this work aims to provide. To address these issues, this study presents a new interpretable deep learning model based on state-of-the-art CNN architectures and incorporating XAI methods such as Grad-CAM++ to explore how the DL model makes decisions.

The key contributions of this research are as follows:

- Development of an Explainable Model: This research proposes a hybrid CNN model for the classification of mosquito species and larvae, incorporating state-of-the-art XAI techniques such as Grad-CAM++ and Score-CAM to enhance transparency and interpretability.
- Evaluation on Multiple Datasets: Unlike previous studies that rely on a single dataset, this study evaluates the proposed model on two distinct datasets, one focused on mosquito species images and the other on mosquito larvae, to validate the model's robustness and generalizability.
- Comprehensive Performance Metrics: The model's performance is evaluated using standard metrics accuracy, precision, recall, and F1-score and compared with existing state-of-the-art approaches.
- Practical Impact: By enhancing the interpretability and performance of mosquito classification models, this research has significant implications for improving public health surveillance and mosquito control efforts.

The rest of the paper is organized as follows: In Section 2, we review recent literature on classifying mosquito species and larvae using deep learning techniques. Section 3 presents the methodology,dataset details, the proposed model architecture, and the evaluation metrics and the Explainable AI. Results and discussion are presented in Section 4. Lastly, Section 5 concludes the paper by summarizing the results and outlining directions for further research.

2 Literature Review

Rapid identification of mosquito species is essential to prevent the transmission of mosquito-borne diseases. In recent years, advanced models such as deep learning approaches, especially Convolutional Neural Networks (CNNs), have demonstrated the ability to automate mosquito species identification, achieving better results by improving both performance quality and speed in this critical task.

A significant study by Goodwin et al. [12] employed CNNs to classify mosquito species, achieving 97.04% accuracy across 16 species. Alternatively, a novelty detection model was also used, achieving 89.07% accuracy when tested on never-seen-before species. The model relied on a library of more than 2,600 mosquito specimens from 67 species, and its tiered approach leveraged the CNN's ability to first extract key information from images before deciding whether the species was known or needed to be introduced anew. This is particularly useful for regions where extensive image databases are not available, as it does not depend on fine-grained data per location. In another study, Lee et al. [16] took a different approach, using color and fluorescence images of mosquitoes combined with the faster R-CNN model. By performing data augmentation (rotating and shifting images), the model is trained to become more robust. This method achieved an F1-score of 97.1%, indicating that deep learning significantly enhanced mosquito classification, particularly relevant for field sites where image quality may be variable and degrade performance. Similarly, Sauer et al. [22] introduced a novel approach of using mosquito wing images instead of full-body images. That was especially useful since the wings are often the most well-preserved parts of a mosquito, while other body regions are broken and shed. After training a CNN model on wing images of seven Aedes species, the accuracy for distinguishing these Aedes mosquitoes from others was 99%. This more straightforward approach for wing images can be particularly valuable when mosquitoes are too damaged to use other identification methods. Similarly, Cannet et al. (2023) performed mosquito species classification using Wing Interferential Patterns (WIPs). WIPs, generated from mosquito wing prints, were a consistent character for species identification in cases where the mosquito body was damaged or hard to distinguish. The deep-learning model developed to analyze these ALV patterns achieved 95% accuracy at the genus level and >85% at the subgenus level. This is a robust species-determination method, particularly when the mosquito body is unavailable for examination.

However, even though promising results were obtained with CNNs, some issues persist, particularly closely related species identification and the treatment of damaged specimens. In Surya et al. [23], Vision Transformers (ViTs) were proposed as a a substitutefor for CNNs for classifying mosquito larvae. They compared ViTs with CNNs, such as ResNet, and showed that ViTs perform better on large-scale datasets. ViTs achieve notable accuracy improvements by using self-attention to capture both local and long-range dependencies, thereby addressing some challenges observed in CNNs. Adhane et al. [1] used citizen science data obtained through the Mosquito Alert project, in which scientists generated a large, heterogeneous dataset of mosquito images contributed by volunteers. This project demonstrated that deep learning models can be helpful for real-world applications, such as monitoring mosquito populations across multiple locations. It highlights the need to incorporate big data from crowd sourcing into large-scale mosquito surveillance and control. A summary of recent related studies is presented in Table 1.

Table 1. Summary of recent related studies.

Ref.	**Dataset**	**Used Model**	**Performance**	**Limitations**
[22]	2696 specimens from 67 species	CNN, Ensemble model	97.04% for closed-set classification, 89.07% for novelty detection	Misclassification within genus, poor performance on rare species
[16]	Images of 11 mosquito species from Korea	Faster R-CNN	Accuracy 97.1%	Challenges distinguishing between species of the same genus
[22]	Wing images from Aedes species	Shallow CNN	99% for Aedes vs. non-Aedes classification	Difficulty in classifying closely related species
[4]	Wing Interferential Patterns (WIPs) of Aedes mosquitoes	CNN	95% genus-level accuracy, 85% subgenus-level	Struggled with similar species, lower accuracy on small datasets
[1]	Images from Mosquito Alert project	Transfer learning with VGG16	High accuracy for Aedes albopictus identification	Dependent on high-quality images
[19]	Images of mosquito larvae for classification	ResNet, ConvNeXT	Best performance with ConvNeXT, high precision and recall	Lower accuracy for smaller datasets, model performance dependent on image quality
[23]	Images of mosquito larvae	ViT, CNN	Better performance with ViTs over CNNs	Larger datasets required for optimal performance, challenges with real-time classification
[5]	Images of mosquito species from Thailand	BYOL-based self-supervised learning	96.77% accuracy, AUC $>$ 99.55%	Requires expert supervision for fine-tuning, data-dependent
[21]	Images of mosquito larvae collected for detection	CNN	90.18% testing accuracy, 92.2% precision	Image quality and larvae size affect accuracy

3 Methodology

An overview of the proposed methodology is shown in Fig. 1.

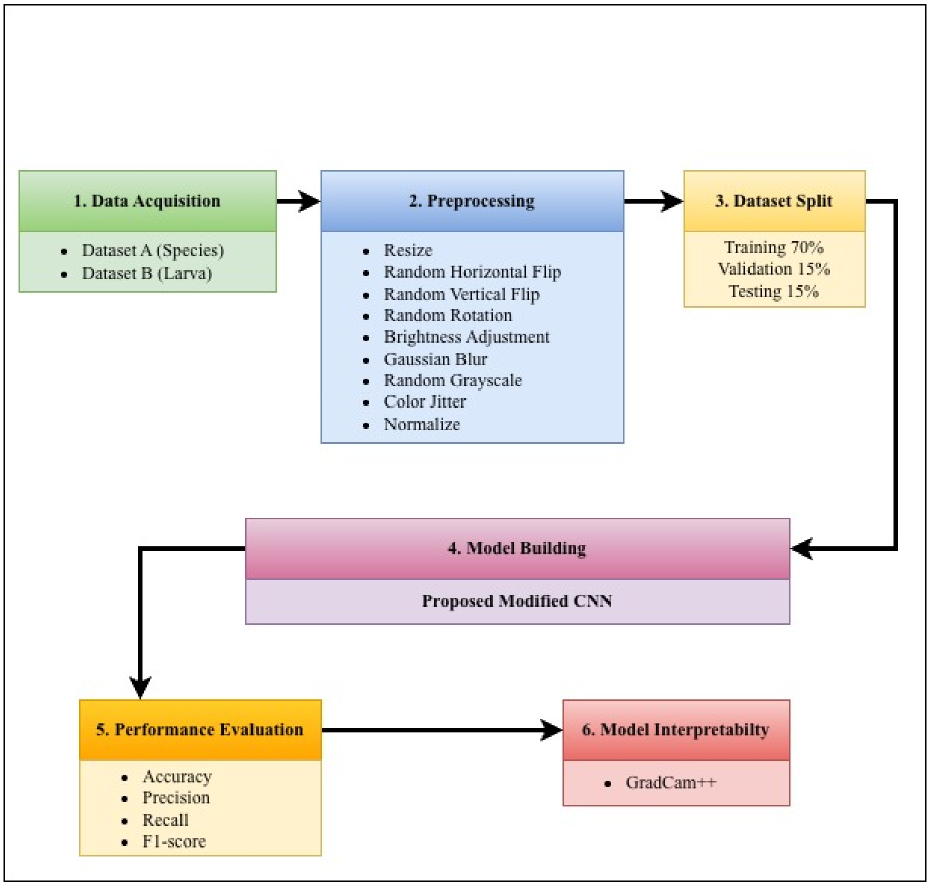

Fig. 1. An overview of the proposed methodology.

3.1 Data Acquisition and Pre-processing

In this study, we used two data sets downloaded from Kaggle: Dataset A[17] and Dataset B[7]. Dataset A consists of 3000 images of three mosquito species (Culex, Anopheles, and Aedes). An example image from both datasets is provided in Fig. 3.1. Dataset B contains four classes (AB, AG, AN, and CQ) of mosquito larvae, with a total of 7463 images. A summary of the datasets is given in Table 2.

The resolution of every image was resized to (256 x 256) pixels to keep the consistency and facilitate model training. Processed images were scaled and aug-

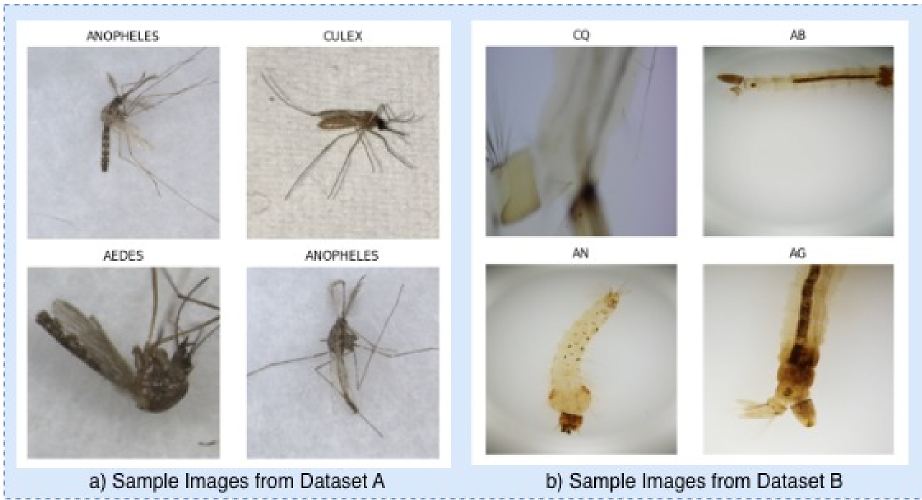

Fig. 2. Sample images from both datasets.

mented by rotating randomly up to 30°, horizontal and vertical reflections, and random affine transformations. The noise was simulated by applying a Gaussian blur (kernel size = 3x3 and sigma in the range 0.1-2.0). In addition, a 10% probability of randomly converting an image to grayscale was also implemented. Furthermore, color jitter was added by scaling brightness and contrast by 0.2 for all images. Finally, all images were transformed into tensors for feeding into the neural network. This pre-processing enhanced the diversity of the training data, facilitating deeper models that learned better generalization. Both datasets were split after pre-processing into training, validation, and test sets with a 70:15:15 split.

3.2 Proposed Model Architecture

The model structure is a variant of a convolutional neural network (CNN) adapted from MobileNetV2, a lightweight deep learning model for efficient computation [13]. This architecture is well-suited to classification tasks with limited computational resources. The model consists of two main parts: feature extraction with MobileNetV2 and final classification with a custom FC classifier. This structure ensures both high classification accuracy and computational efficiency. In this section, we provide an in-depth description of the design of our model, from its backbone architecture to its output layers. Layer architecture of the proposed CNN model is presented in Fig. 3.

Base Model (MobileNetV2) The backbone architecture of the proposed model is MobileNetV2, an efficient CNN that uses depthwise separable convolutions to reduce parameter count and computational cost. This architecture is

Table 2. Description of the Datasets

Dataset Name	Class Name	Number of Images	Total
Dataset A [17]	Mosquito Species Image Dataset		3000
	Culex	1000	
	Anopheles	1000	
	Aedes	1000	
Dataset B [7]	MosquitoLarvae-7400		7463
	AB (Aedes albopictus)	1932	
	AG (Aedes aegypti)	1843	
	AN (Anopheles cracens)	1852	
	CQ (Culex quinquefasciatus)	1836	

optimized for the mobile and embedded systems space, providing a compromise between performance and power efficiency.

The feature extractor in this model architecture is MobileNetV2. The pre-trained MobileNetV2 model, trained on the ImageNet dataset, is adopted to extract hierarchical representations from input images. We remove the original MobileNetV2 classifier and train only on the convolutional base for feature generation, so that our model can learn better representations from input images by leveraging prior knowledge from ImageNet.

The pre-trained weights of MobileNetV2 provide rich, high-level features from the input image, which are used as input to the next part of the model for final classification.

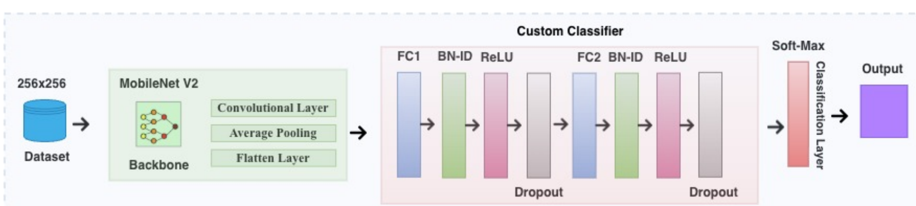

Fig. 3. Layer architecture of the proposed CNN model.

Custom Fully Connected Classifier The second part of the architecture is a custom fully connected classifier that takes the features extracted from MobileNetV2 and provides the final classification. We build the classifier using two fully connected layers along with Batch Normalization and ReLU activation, and a softmax-activated output layer. Fine-tuning is enabled by the custom classifier, which receives pre-trained feature representations from MobileNetV2. Hyperparameters for the proposed model is displayed in Table 3.

Table 3. Hyperparameters for the proposed model.

Hyperparameter	Value	Description
Batch Size	32	Number of samples processed in one forward/backward pass.
Learning Rate	0.001	Step size at each iteration while moving toward a minimum of the loss function.
Optimizer	Adam	Optimizer used for updating the model weights during training.
Dropout Rate	0.5	Fraction of the input units to drop during training to prevent overfitting.
Activation Function	ReLU	Activation function used to introduce non-linearity into the model.
Loss Function	Cross Entropy Loss	Function used to calculate the error between the predicted and actual values.
Epochs	20	Number of complete passes through the training dataset.

First Fully Connected Layer: The first FC layer takes the feature vector F_{extract} and maps it to a 256-dimensional space. This is followed by Batch Normalization, which normalizes the activations to assist with training. The result of this layer is fed into a ReLU Activation function, which introduces nonlinearity to the data, allowing our systems to learn complex patterns. The operation in this layer can be expressed as:

$$F_{\text{extract}} = \text{MobileNetV2}(x) \tag{1}$$

Where:

$$x \in \mathbb{R}^{H\times W\times 3}$$

is the input image of size $H \times W \times 3$ (height H, width W, and 3 color channels).

$$F_{\text{extract}} \in \mathbb{R}^{d}$$

is the output feature vector, where d is the number of extracted features.

Second Fully Connected Layer: We apply another 128-unit fully connected layer to the h_1 output by passing it through the second fully connected layer. Similar to the first layer, we use Batch Normalization and ReLU activation to stabilize training and enable complex decision boundaries. The operation is expressed as:

$$h_2 = \text{ReLU}\left(\text{BatchNorm}\left(W_2 h_1 + b_2\right)\right) \tag{2}$$

Where:

- $W_2 \in \mathbb{R}^{128\times 256}$ is the weight matrix for the second fully connected layer.
- $b_2 \in \mathbb{R}^{128}$ is the bias vector for the second layer.
- $h_2 \in \mathbb{R}^{128}$ is the output of the second layer after applying ReLU activation.

Output Layer The final layer is the output layer, which predicts the class probabilities. The output of the second fully connected layer h_2 is passed through a final fully connected layer, which reduces the feature dimension to C, where C is the number of classes in the classification task. This is followed by the softmax activation function, which converts the raw output (logits) into a probability distribution. The class with the highest probability is selected as the predicted class. The operation is given by:

$$y_{\text{pred}} = \text{Softmax}(W_3 h_2 + b_3) \tag{3}$$

Where:

- $W_3 \in \mathbb{R}^{C \times 128}$ is the weight matrix for the output layer, with C being the number of classes.
- $b_3 \in \mathbb{R}^C$ is the bias vector for the output layer.
- $y_{\text{pred}} \in \mathbb{R}^C$ is the predicted probability distribution for each class.

The softmax function is defined as:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \tag{4}$$

Where:

- z_i represents the logits (raw output values) for class i.
- The softmax function converts these logits into a probability distribution, where the predicted class is the one with the highest probability.

3.3 Performance Evaluation Metrics

In this study, accuracy, precision, recall, and F1-score were used as performance measures to evaluate the classification potential of the proposed model. Below are the definitions and the mathematical formulations for each metric:

Accuracy Accuracy is the proportion of correct predictions (both true positives and true negatives) out of the total predictions [3]. It measures the overall performance of the model.

Mathematical Formula:

$$\text{Accuracy} = \frac{T_p + T_n}{\text{Total Sample}} \tag{5}$$

Where:

- T_p is the number of true positives.
- T_n is the number of true negatives.
- Total Sample is the total number of samples in the dataset.

Precision Precision is the proportion of true positive predictions out of all the positive predictions made by the model [24]. It indicates how many of the predicted positive instances were actually positive.

Mathematical Formula:

$$\text{Precision} = \frac{T_p}{T_p + F_p} \tag{6}$$

Where:

- T_p is the number of true positives.
- F_p is the number of false positives.

Recall Recall (also known as Sensitivity or True Positive Rate) is the proportion of true positive predictions out of all the actual positive instances in the dataset [18]. It indicates how well the model detects positive instances.

Mathematical Formula:

$$\text{Recall} = \frac{T_p}{T_p + F_n} \tag{7}$$

Where:

- T_p is the number of true positives.
- F_n is the number of false negatives.

F1-score The F1-score is the harmonic mean of precision and recall, providing a single metric that balances both [8]. It is especially useful when dealing with imbalanced classes.

Mathematical Formula:

$$F1\text{-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

3.4 Explainable AI Interpretable Techniques

GradCam++ Grad-CAM++ is an advanced explainable AI technique designed to interpret and visualize the outcomes of deep learning models, particularly for image classification tasks [11]. It improves upon the original Grad-CAM by offering more detailed localization information, pinpointing the regions in the input image that contributed to the model's final decision. While Grad-CAM assumes a linear contribution of each pixel and performs global average pooling of gradients, Grad-CAM++ employs weighted sums of higher-order derivatives, enabling it to capture more nuanced details about multiple regions of the same class within the image. This enhancement enables Grad-CAM++ to generate sharper, more precise attention maps [6], making it especially valuable for tasks such as Image classification, object detection, and other complex visual-based challenges. Mathematically, it is expressed as:

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial^2 Y^c}{\partial A_k^2} \tag{9}$$

4 Result and Discussion

4.1 Experimental Settings

The experiments were conducted on a Kaggle T4 GPU. The modified deep learning model was developed in PyTorch (version 2.2.1) to meet the specific requirements of the model. Furthermore, it used some important libraries, namely NumPy and Matplotlib, which were integrated into the experimental environment.

4.2 Performance Metrics Analysis

Table 4. Performance Metrics Summary

Phase	Accuracy	Loss	Precision	Recall	F1-score
Dataset A (Mosquito Species)					
Training	0.9967	0.5567	0.9967	0.9967	0.9967
Validation	0.9844	0.5657	0.9848	0.9844	0.9844
Testing	0.9956	0.5577	0.9956	0.9956	0.9956
Dataset B (Mosquito Larvae)					
Training	0.8316	0.9120	0.8316	0.8305	0.8393
Validation	0.8382	0.9076	0.8316	0.8305	0.8355
Testing	0.8295	0.9144	0.8336	0.8295	0.8269

The modified deep learning model proposed here demonstrated overall excellence in classifying mosquito species and larvae. On Dataset A (Mosquito Species), the model's training accuracy was 99.67%, whereas on Dataset B (Mosquito Larvae), it reached 83.16% after 10 epochs. The performance results during training, validation and testing are shown in Table 4. These findings indicate that the model has learned the complex patterns of data well, with good generalization to new data. Significantly, the model achieved test accuracies of 99.56% for Dataset A and 82.95% for Dataset B, demonstrating robustness in predicting mosquito species and larvae on test samples not seen during training. The continual decrease in training and validation losses shows that the learning process is continuing on an improving track. It implies that the model can be trained without overfitting and can also generalize when applied in practice.

4.3 Accuracy and Loss Curve Analysis

The accuracy and loss curves visualize the model's performance over training epochs. The accuracy curve shows the model's classification performance. The loss curve shows the trend in error over time for models during training. Fig.

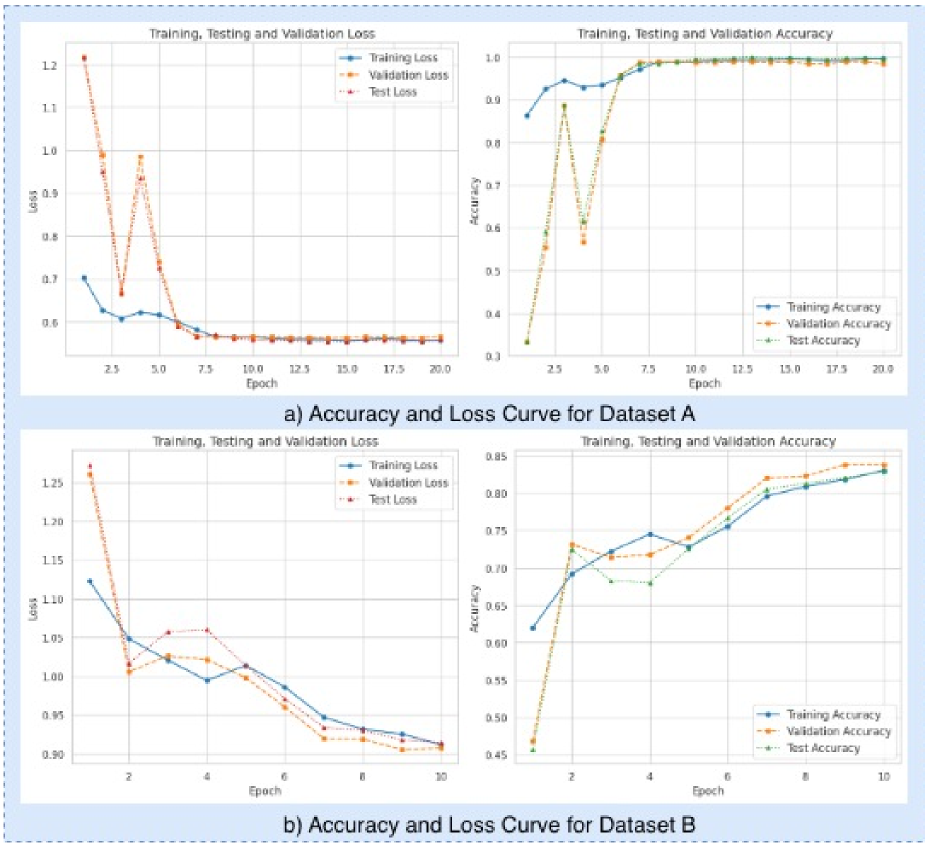

Fig. 4. Accuracy and loss curves for both datasets.

4 shows the training and validation accuracy and loss curves for Dataset A (Mosquito Species) and Dataset B (Mosquito Larvae).

It can be observed from the results that, for both datasets, the loss decreases rapidly in the early epochs and then becomes almost constant. For Dataset A, the loss decreases rapidly after epoch two and then converges after epoch 7. Meanwhile, accuracy increases after the second epoch and finally stabilizes at a saturating level after 10 epochs. The curves of accuracy during training and validation are almost identical, indicating that the two trends correspond well, with a small training gap. Training and validation losses adjust very well. The validation loss increases slightly, but without any dramatic changes during fitting (indicating no overfitting), as the number of iterations increases.

For Dataset B, the training and validation losses decrease considerably and stabilize by the 10th epoch, while, as in Dataset A, the accuracy increases uniformly over this period. As in Dataset A, the training and validation accuracy curves already showed similar behavior, and the model kept improving until reaching a point that suggests it continues to learn well with no apparent underfit.

4.4 Confusion Matrix

Fig. 5 presents the confusion matrices for multiclass classification of mosquito species and larvae using the proposed modified deep learning model. The diagonal entries correspond to accurate class representations, and off-diagonal entries are indicative of a wrong classification.

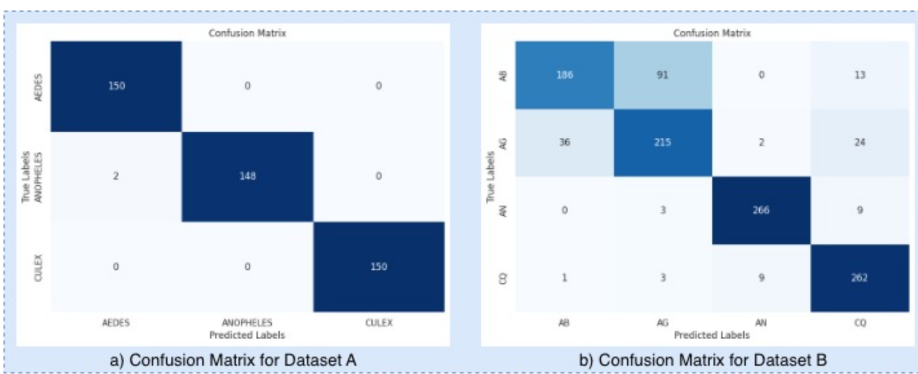

Fig. 5. Confusion matrix analysis for both datasets.

For Dataset A (Mosquito Species) (Fig. 5a), the model displayed an outstanding performance in classification, as shown by 150 Aedes, 148 Anopheles, and 150 Culex samples being correctly classified, with only two Aedes misclassified as Anopheles, and two Anopheles misclassified as Aedes.

For Dataset B (Mosquito Larvae) (Fig. 5b), further indicating the quality of the model: 186 AB, 215 AG, 266 AN and 262 CQ samples were accurately predicted. Misclassifications consisted of 36 AB predicted to be AG, 3 AG predicted to be AN, and 1 AN predicted to be AB. 3 CQ were also incorrectly predicted to be AN, while the remaining predictions were correct. These efforts demonstrate the model's robust capacity to differentiate species or larvae, with only a few misclassifications, reaffirming its high predictive accuracy.

4.5 Model Interpretability Analysis

In this work, Grad-CAM++ was used to interpret the decision of the proposed classification model for mosquito species and larvae. Figs. 6 and 7 show the Grad-CAM++ visualizations for each dataset. The first column of all figures shows the original images, and Grad-CAM++ heatmaps are superimposed on them in the second column.

For Dataset A (Mosquito Species), Fig. 6a shows that the Grad-CAM++ heatmap focuses on areas such as the body and wings of mosquitoes, with warmer (or red) regions representing the most contributive locations in the image to classification. For Aedes, the heatmap shows a trend across some areas of the wings and body image, indicating that these features play an important role in the model's species classification. Likewise, for Anopheles and Culex, the Grad-CAM++ visualizations show where it is important to discriminate against these mosquitoes, and it becomes apparent that the model is focused on specific structures of mosquitoes.

For Dataset B (Mosquito Larvae), Fig. 7, the Grad-CAM++ maps also indicate that our model focuses on the important areas of the larvae's bodies. For AB, AG, AN, and CQ larvae, heatmaps indicate distinct body segments, with red pixels indicating key features that contributed to the model's decision. Such visualizations offer interpretable insights into which image features were most effective for predicting and help us understand how the model makes its decisions.

Altogether, the Grad-CAM++ visualizations do a good job of exposing image regions that the model relies on to differentiate species and larvae, and further confirm that it provides transparent and trustworthy predictions.

4.6 Discussion

In this research, we design a deep learning model with MobileNetV2 as the backbone architecture for mosquito species and mosquito larvae classification. We tested our model with 99.67% accuracy on Mosquito Species (Dataset A) and 83.16% on Mosquito Larvae (Dataset B), demonstrating broader classification capabilities not only for species but also for larvae, showing the robustness of our model for classifying mosquito species and larvae.

As demonstrated in Table 5, the proposed MobileNetV2 achieves better performance than several recent state-of-the-art models in captioning ability for

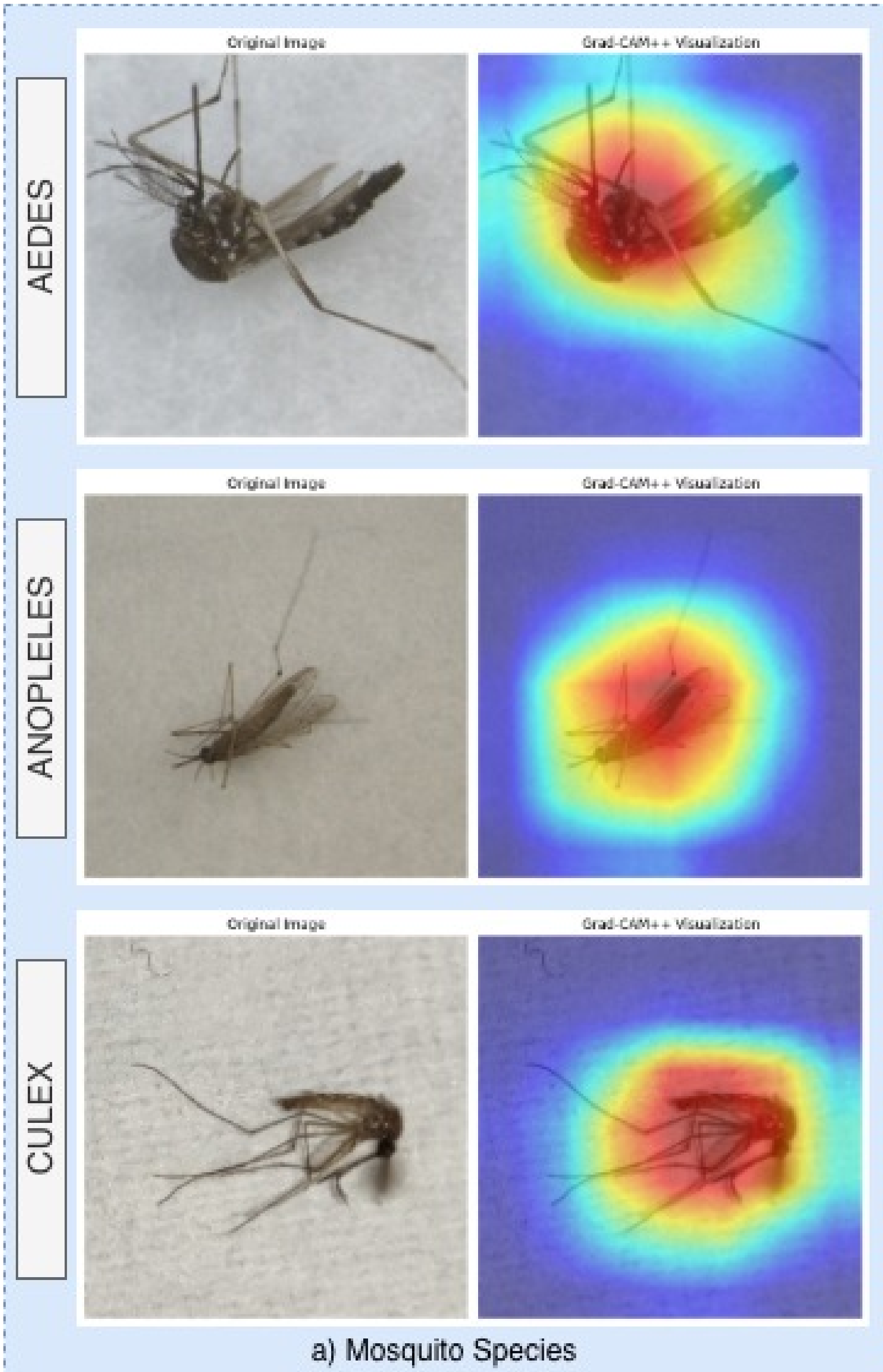

Fig. 6. GradCam++ visualization for mosquito species

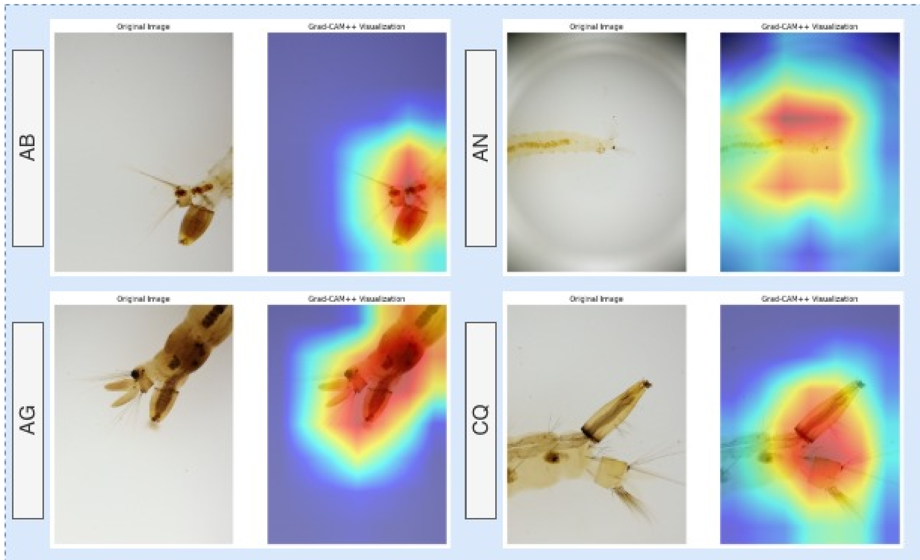

Fig. 7. GradCam++ visualization for mosquito larvae

mosquito species and larvae. Interpretability is important for AI-driven biological classification; accordingly, the study introduced Grad-CAM++ to explain the model. The importance maps of the images that affected the model's decision-making were visualized using Grad-CAM++. The Grad-CAM heatmaps provide an intuitive explanation of which areas of the mosquito species and larvae images the model focuses on for prediction, helping experts locate meaningful features for mosquito identification or larval classification.

This way, not only can the accuracy be high, but also model decision-making commensurate with biological conditions can be made interpretable. Grad-CAM++ visualizes regions that contribute to the decision and correlate with quantitatively important mosquito species and larval features, thereby confirming that the model can be applied to practical problems such as entomological pest control and vector-borne disease management.

5 Conclusion and Future works

In this work, we present a deep learning model based on MobileNetV2 to classify both mosquito species and larvae, achieving 99.67% accuracy on Dataset-A (Mosquito Species) and 83.16% on Dataset-B (Mosquito Larvae). Integrating Grad-CAM++ makes the model more interpretable, since it shows which parts of the input contribute to the prediction. The proposed model achieves high predictive performance and interpretability, providing a practical aid for mosquito monitoring and disease defense.

For future work, investigating and leveraging efficient CNN models to improve computational efficiency, as well as applying model compression methods such as pruning, quantization, and knowledge distillation for deployment on edge devices. Furthermore, extending the classifier to distinguish more mosquito species and larval instars would make it more suitable for public health surveillance.

6 GitHub

https://github.com/TahmidEnam/Data-Warehousing-and-Data-Mining

References

1. Adhane, G., Dehshibi, M.M., Masip, D.: A deep convolutional neural network for classification of aedes albopictus mosquitoes. IEEE Access **9**, 72681–72690 (2021). https://doi.org/10.1109/ACCESS.2021.3079700
2. Amiruddin, B.P., Kadir, R.E.A.: Cnn architectures performance evaluation for image classification of mosquito in indonesia. In: 2020 International Seminar on Intelligent Technology and Its Applications (ISITIA). pp. 223–227. IEEE (2020)
3. Cabot, J.H., Ross, E.G.: Evaluating prediction model performance. Surgery **174**(3), 723–726 (2023)
4. Cannet, A., Simon-Chane, C., Histace, A., Akhoundi, M., Romain, O., Souchaud, M., Sereno, D.: Wing interferential patterns (wips) and machine learning for the classification of some aedes species of medical interest. Scientific Reports **13**(1), 17628 (2023)
5. Charoenpanyakul, R., Kittichai, V., Eiamsamang, S., Sriwichai, P., Pinetsuksai, N., Naing, K.M., Chuwongin, S.: Enhancing mosquito classification through self-supervised learning. Scientific Reports **14**(1), 27123 (2024)
6. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 839–847. IEEE (Mar 2018)
7. Cyberthorn: Mosquitolarvae-7400 classification. https://www.kaggle.com/datasets/cyberthorn/mosquitolarvae-7400-classification (2023)
8. Diallo, R., Edalo, C., Awe, O.O.: Machine learning evaluation of imbalanced health data: a comparative analysis of balanced accuracy, mcc, and f1 score. In: Practical Statistical Learning and Data Science Methods: Case Studies from LISA 2020 Global Network, USA, pp. 283–312. Springer Nature Switzerland (2024)
9. Fariza, A., Yuwono, W., Akbar, R., Asmara, R., Aryawan, I.G.K.P.: Mobile based mosquito larvae recognition from photo image using convolutional neural network. Advances in Science and Technology **126**, 128–136 (2023)
10. Fernandes, M.S., Cordeiro, W., Recamonde-Mendoza, M.: Detecting aedes aegypti mosquitoes through audio classification with convolutional neural networks. Computers in biology and medicine **129**, 104152 (2021)
11. George, A.B., Bathini, S.: Grad-cam & grad-cam++ for explainable oral squamous cell carcinoma detection using deep learning on orthopantomograms. In: 2025 International Conference on Sensors and Related Networks (SENNET) Special Focus on Digital Healthcare. pp. 1–6. IEEE (Jul 2025)

12. Goodwin, A., Padmanabhan, S., Hira, S., Glancey, M., Slinowsky, M., Immidisetti, R., Acharya, S.: Mosquito species identification using convolutional neural networks with a multitiered ensemble model for novel species detection. Scientific Reports **11**(1), 13656 (2021)
13. Hamid, Y., Wani, S., Soomro, A.B., Alwan, A.A., Gulzar, Y.: Smart seed classification system based on mobilenetv2 architecture. In: 2022 2nd international conference on computing and information technology (iccit). pp. 217–222. IEEE (2022)
14. Hossain, M.S., Raihan, M.E., Hossain, M.S., Syeed, M.M., Rashid, H., Reza, M.S.: Aedes larva detection using ensemble learning to prevent dengue endemic. BioMedInformatics **2**(3), 405–423 (2022)
15. Kittichai, V., Kaewthamasorn, M., Samung, Y., Jomtarak, R., Naing, K.M., Tongloy, T., Boonsang, S.: Automatic identification of medically important mosquitoes using embedded learning approach-based image-retrieval system. Scientific Reports **13**(1), 10609 (2023)
16. Lee, S., Kim, H., Cho, B.K.: Deep learning-based image classification for major mosquito species inhabiting korea. Insects **14**(6), 526 (2023)
17. Masud, M.: Mosquito dataset for classification cnn. https://www.kaggle.com/datasets/masud1901/mosquito-dataset-for-classification-cnn (2023)
18. Miao, J., Zhu, W.: Precision–recall curve (prc) classification trees. Evolutionary Intelligence **15**(3), 1545–1569 (2022)
19. Nainggolan, P.I., Efendi, S., Budiman, M.A., Lydia, M.S., Rahmat, R.F., Bukit, D.S., Sulaiman, R.: Detection and classification of mosquito larvae based on deep learning approach. Engineering Letters **33**(1) (2025)
20. Park, J., Kim, D.I., Choi, B., Kang, W., Kwon, H.W.: Classification and morphological analysis of vector mosquitoes using deep convolutional neural networks. Scientific reports **10**(1), 1012 (2020)
21. Saeed, M.S., Nazreen, S.F., Ullah, S.S.S.A., Rinku, Z.F., Rahman, M.A.: Detection of mosquito larvae using convolutional neural network. In: 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST). pp. 478–482. IEEE (2021)
22. Sauer, F.G., Werny, M., Nolte, K., Villacañas de Castro, C., Becker, N., Kiel, E., Lühken, R.: A convolutional neural network to identify mosquito species (diptera: Culicidae) of the genus aedes by wing images. Scientific Reports **14**(1), 3094 (2024)
23. Surya, A., Peral, D.B., VanLoon, A., Rajesh, A.: A mosquito is worth 16x16 larvae: Evaluation of deep learning architectures for mosquito larvae classification. arXiv preprint arXiv:2209.07718 (2022)
24. Varoquaux, G., Colliot, O.: Evaluating machine learning models and their diagnostic value. In: Machine learning for brain disorders, pp. 601–630 (2023)
25. Vega Rúa, A., Okech, B.A.: The spread of mosquito-borne diseases: a major and global public health problem. In: Olfactory Concepts of Insect Control-Alternative to Insecticides: Volume 1, pp. 1–27. Springer International Publishing, Cham (2019)
26. Yu, K., Wu, J., Wang, M., Cai, Y., Zhu, M., Yao, S., Zhou, Y.: Using uav images and deep learning in investigating potential breeding sites of aedes albopictus. Acta Tropica **255**, 107234 (2024)

Table 5. Summary of Recent Studies on Mosquito Classification.

Ref.	**Dataset**	**Model**	**Performance**	**Limitations**
[22]	2696 samples from 67 species	CNN, Ensemble	97.04% (closed-set), 89.07% (novelty detection)	Misclassification within genus, poor performance on rare species
[16]	11 mosquito species images from Korea	Faster R-CNN	Accuracy: 97.1%	Difficulty distinguishing species within same genus
[22]	Wing images from Aedes species	Shallow CNN	99% for Aedes vs. non-Aedes classification	Challenges classifying closely related species
[4]	Wing Interferential Patterns (WIPs) of Aedes mosquitoes	CNN	95% genus, 85% subgenus-level	Struggled with similar species, low accuracy on small datasets
[1]	Mosquito Alert project images	VGG16 (Transfer Learning)	High accuracy for Aedes albopictus identification	Dependent on high-quality images
[19]	Mosquito larvae images for classification	ResNet, ConvNeXT	Best performance with ConvNeXT, high precision and recall	Lower accuracy on smaller datasets, image quality dependency
[23]	Mosquito larvae images	ViT, CNN	ViT outperforming CNNs	Larger datasets needed, real-time classification challenges
[5]	Mosquito species images from Thailand	BYOL-based self-supervised learning	96.77% accuracy, AUC > 99.55%	Needs expert supervision, data-dependent
[21]	Mosquito larvae images for detection	CNN	90.18% testing accuracy, 92.2% precision	Accuracy affected by image quality and larvae size
This Study	Mosquito Species (Aedes, Anopheles, Culex)	MobileNetV2 + Custom Classifier	Training Acc: 99.67%, Test Acc: 99.56% (Species); Training Acc: 83.16%, Test Acc: 82.95% (Larvae)	Lower performance on larvae classification, more data needed for better generalization