A Lightweight Deep Learning Pipeline for Real-Time Identification of Mosquito Larvae on Mobile Devices

Mohammad Fadly Syahputra[1*], Opim Salim Sitompul[1], Fahmi[2], Maya Silvi Lydia[3], Abel Agustian Sidauruk[3], Syahril Efendi[3], Pauzi Ibrahim Nainggolan[4], Dhani Syahputra Bukit[5], Riza Sulaiman[6]

[1] Department of Information Technology, Universitas Sumatera Utara, Medan, 20155, Indonesia
[2] Department of Electrical Engineering, Universitas Sumatera Utara, Medan, 20155, Indonesia
[3] Department of Computer Science, Universitas Sumatera Utara, Medan, 20155, Indonesia
[4] Computer Vision and Multimedia Laboratory, Universitas Sumatera Utara, Medan, 20155, Indonesia
[5] Department of Public Health, Universitas Sumatera Utara, Medan, 20155, Indonesia
[6] Institute of Visual Informatics, Universiti Kebangsaan Malaysia, Bangi, 43600 Malaysia
E-mail: nca.fadly@usu.ac.id
[*]Corresponding author

Keywords: detection, classification, MobileNetV3, YOLOv8, mosquito larvae

Received: September 6, 2025

Real-time detection and classification of mosquito larvae on mobile devices still face challenges in terms of accuracy and efficiency. Manual identification is limited, thereby necessitating the development of deep learning-based systems to improve both accuracy and speed in diagnosis. This study proposes a fusion deep learning model, combining YOLOv8 for object detection and MobileNetV3-Small for mosquito larvae classification, to enhance the accuracy and efficiency of classifying mosquito larvae into three classes—Aedes, Culex, and an 'Unknown' class that captures non-Aedes/Culex larvae (e.g., Anopheles, Toxorhynchites)—as well as to support environmental health monitoring. The methodology involves using YOLOv8 for object detection and MobileNetV3 for mosquito larvae classification. The dataset comprises images of Aedes and Culex larvae and curated "Unknown" examples representing other genera. The model was trained and evaluated using deep learning techniques, and subsequently deployed in a mobile application to automatically detect and classify the larvae. The results indicate that the developed system is capable of detecting and classifying mosquito larvae with high accuracy, with YOLOv8 achieving mAP@0.5 of 0.986 and mAP@0.5:0.95 of 0.777, while MobileNetV3-Small attained a classification accuracy of 0.962. For efficiency, the model runs in real time on mobile devices with low latency. The model also demonstrates stable performance on unseen data, confirming its potential for environmental health monitoring and its role in supporting more effective vector control efforts, as well as contributing to further research in the field of entomology.

Povzetek:

1 Introduction

One of the primary challenges in vector-borne disease control is the lack of a rapid, accurate, and efficient method for identifying mosquito larvae that researchers or health workers can deploy in the field. Diseases such as dengue fever, malaria, chikungunya, Zika, and yellow fever remain serious threats to global public health, with reported annual mortality exceeding 700,000 cases [1][2]. The proximity of mosquito breeding sites to human settlements further elevates the risk of disease transmission [3][4]. Currently, there are no specific treatments or commercially available vaccines for most arboviruses, making mosquito population control the only effective preventive measure [5][6][7].

Vector control efforts have traditionally focused on adult mosquitoes, whereas a more preventive approach involves detecting and eliminating larvae before they reach the infectious stage [8]. Early identification of larvae is a crucial step in interrupting the transmission cycle. However, conventional identification methods rely on visual observation, are highly dependent on individual expertise, are time-consuming, and are prone to human error [9]. Entomological characterization is essential for understanding mosquito species behavior, but current identification practices remain manual and require experienced specialists [10][11][12].

In recent years, advancements in deep learning and computer vision have created new opportunities for automating the detection and classification of organisms, including mosquito larvae [13]. This approach can improve the speed, accuracy, and scalability of vector control programs. Several studies have proposed the use of convolutional neural networks (CNNs) for mosquito

larva classification [14][15]; however, challenges such as high computational demands and hardware limitations remain significant, particularly for field deployment on mobile devices. Therefore, developing more efficient and adaptable automated methods is crucial [11][16]. Given that these diseases continue to burden many tropical countries, vector prevention and control strategies remain the most feasible large-scale intervention for governments to curb the spread of arboviruses [17].

At the national level, Indonesia—a tropical country with over 457 mosquito species from 18 genera—faces a high risk of vector-borne disease transmission, further exacerbated by its humid climate and high rainfall. Three major genera, Aedes, Culex, and Anopheles, dominate disease transmission [18]. Hence, there is an urgent need for a rapid, accurate, and field-accessible larval identification system, notably to support community-based control programs and swift responses to potential outbreaks.

This study proposes the development of a mosquito larva detection and classification system based on deep learning, integrating the YOLOv8 architecture for object detection and MobileNetV3 for species classification. YOLOv8 offers real-time detection capabilities with high accuracy under various environmental conditions [19], while MobileNetV3 is designed for high computational efficiency on resource-constrained devices, such as smartphones. The integration of these two models aims to produce a solution that is not only accurate and reliable but also lightweight and practical for implementation in mobile applications [20]. This research is expected to improve environmental health management through real-time field detection of mosquito larvae.

Table 1: Comparison of related studies in mosquito larvae detection and classification.

Article	Detection Method	Classification	Advantages	Limitations
A YOLO-Based Approach for Aedes Aegypti Larvae Classification and Detection [21]	YOLOv3	Not applied	The system is capable of detecting *Aedes* larvae in real-time on resource-constrained devices with an average of 6.8 FPS and an RMSE of 0.45 for *Aedes* and 0.77 for non-*Aedes*	The system still experiences misclassification, unstable predictions, and is influenced by lighting conditions and the imbalance in the test data
Detection of Mosquito Larvae Using Convolutional Neural Network [8]	Not applied	CNN (Keras + TensorFlow)	The model achieves an accuracy of 93.95% (training), 90.18% (validation), and a precision of 92.2%. It is deployable on a Raspberry Pi and is useful for mosquito larvae detection	The system only detects whether an object is a larva or not, without species detection
An Improved Transfer Learning Based Larvae Detection and Classification Using Densenet121 [22]	Densenet121	Not applied	The model achieves high accuracy and is relevant for larva-based disease detection with support from AI technology and drones	Drone data is still limited, and the model has not been tested in real-world conditions with a wide range of environmental variations
Classification Of Aedes Mosquito Larva Using Convolutional Neural Networks And Extreme Learning Machine [14]	Not applied	CNN And Extreme Learning Machine	The CNN-ELM model with a pre-trained CNN achieves an accuracy of 98%, an F1-score of 99% for *Aedes*, and an F1-score of 96% for non-*Aedes*, providing excellent classification between *Aedes* and non-*Aedes* larvae	The system can only classify two types of larvae (*Aedes* and non-*Aedes*) and does not include other mosquito larva species
Detection and Classification of Mosquito Larvae Based on Deep Learning Approach [15]	YOLOv5, FPSnet	Not applied	The YOLOv5 detection method achieves 97% accuracy with a precision of 94.4%, recall of 95.7%, and an mAP@0.5 of 0.971, delivering better detection results for *Aedes* (99.3%), *Anopheles* (97%), and *Culex* (97.8%)	This study has not effectively measured computational time and resource consumption, and the tasks of detection and classification are separated, requiring significant computational resources

2 Related studies

The detection and classification of mosquito larvae have become a significant area of research due to their relevance in vector-borne disease control and

environmental monitoring. Numerous studies have utilized advanced techniques like deep neural network architectures—particularly YOLOv8 and MobileNetV3—and have investigated their implementation on portable devices such as smartphones to facilitate real-time field detection.

Nevertheless, existing studies face challenges, including more representative datasets, higher larval classification accuracy, and model optimization for efficient operation on resource-constrained devices. Table 1 presents a comparative overview of the approaches used in recent mosquito larva detection and classification studies.

Based on the analysis of Table 1, most prior research has concentrated primarily on larva detection without fully implementing species-level larval classification. Of the five articles reviewed, most employed detection methods based on YOLO variants such as YOLOv3 and YOLOv5, as well as other CNN architectures like DenseNet121 and CNN-ELM. However, only one study integrated a classification method using CNN and an Extreme Learning Machine.

Several studies have demonstrated strong performance in detection accuracy and even succeeded in deploying the models on devices such as Raspberry Pi and drones. However, a standard limitation was the lack of differentiation among larval species (e.g., Aedes, Anopheles, and Culex). This highlights a critical gap between detection and specific species classification of mosquito larvae—an aspect essential to support vector-borne disease control efforts.

As a comparison, a study titled "Detection and Classification of Mosquito Larvae Based on Deep Learning Approach" [15] utilized YOLOv5 and FPSNet, achieving high accuracy in detecting and classifying three larval species. Nevertheless, the study also emphasized that such accuracy required a large dataset and substantial computational resources.

In contrast to previous approaches, the present study employs the YOLOv8 architecture for object detection and MobileNetV3 for classifying mosquito larva species. This combination enables real-time larva detection while efficiently classifying species on mobile devices. MobileNetV3 is chosen for its computational efficiency on low-resource devices, while YOLOv8 provides improved accuracy and speed in detection compared to earlier versions. As such, this approach aims to bridge the gap identified in prior studies by integrating mosquito larvae detection and classification into a single, unified system.

3 Methodology

This study developed a two-stage system to detect and classify mosquito larvae into three operational classes: Aedes, Culex, and an "Unknown" class. The Unknown class is defined as non-Aedes/non-Culex larvae and, in this study, is composed of images from the genera Anopheles and Toxorhynchites. The Unknown class is included to (i) prevent overconfident misassignment of other genera to Aedes/Culex and (ii) better reflect the taxonomic diversity that field users may encounter. Unless otherwise noted, the same preprocessing and augmentation pipeline was applied to all three classes to reduce any source-dependent bias introduced by differing image repositories.

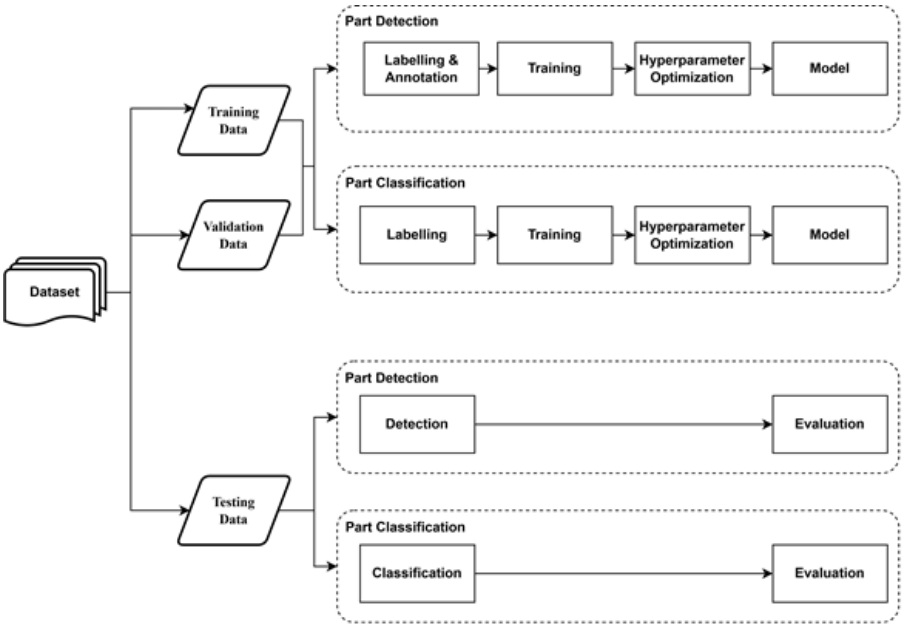


Figure 1: Methodology

The workflow of this study begins with the dataset preparation process. The dataset is divided into three subsets: training data, validation data, and testing data. The training and validation sets are used to build and optimize two separate models: the detection model and the

classification model. The detection model's training data undergo a labeling and annotation process before being used for model training and hyperparameter tuning. Meanwhile, the classification model is trained and fine-tuned directly using the training data.

Once both models are fully trained, the testing phase is carried out using the testing dataset. In this phase, the detection model is first applied to identify larvae in the test images, and the detection results are evaluated. Each detected larva is cropped based on its bounding box and passed to the classification model to determine its genus. The classification results are then evaluated using standard performance metrics such as accuracy, precision, and recall. The complete research workflow is illustrated in Figure 1.

3.1 Data acquisition

The dataset used in this study was obtained from two primary sources. Images for the Aedes and Culex classes were collected in-house at the Environmental Health Engineering and Disease Control Center (BTKLPP) Class 1 Medan. To build a third class that captures non-Aedes/non-Culex larvae encountered in practice, we constructed an "Unknown" class composed of two genera—Anopheles and Toxorhynchites—sourced from publicly available repositories. After preprocessing and quality control, each class contained 1,460 images. For detection, images were resized to 640×640 and annotated; for classification, cropped detections were resized to 360×360 (see 3.2 for preprocessing details). Figure 2 presents sample images from the dataset used in this study.

Figure 2: Dataset collection

In this study, the "Unknown" class is explicitly defined as non-Aedes/non-Culex larvae and consists of images from two genera: Anopheles and Toxorhynchites. This design reduces over-confident misassignment of other genera to Aedes/Culex and better reflects field variability.

Anopheles images were obtained from the Roboflow Universe "Mosquito Larvae Dataset" project [27] (project page cited in References). This repository aggregates labeled larval imagery suitable for computer-vision training. For Toxorhynchites, no single consolidated public larval dataset was identified. Accordingly, we curated images from openly accessible web pages and educational videos discovered via generic web search (e.g., Google Images/YouTube). Because these materials come from heterogeneous hosts and licenses, per-image URLs were not consistently retained during early exploratory curation.

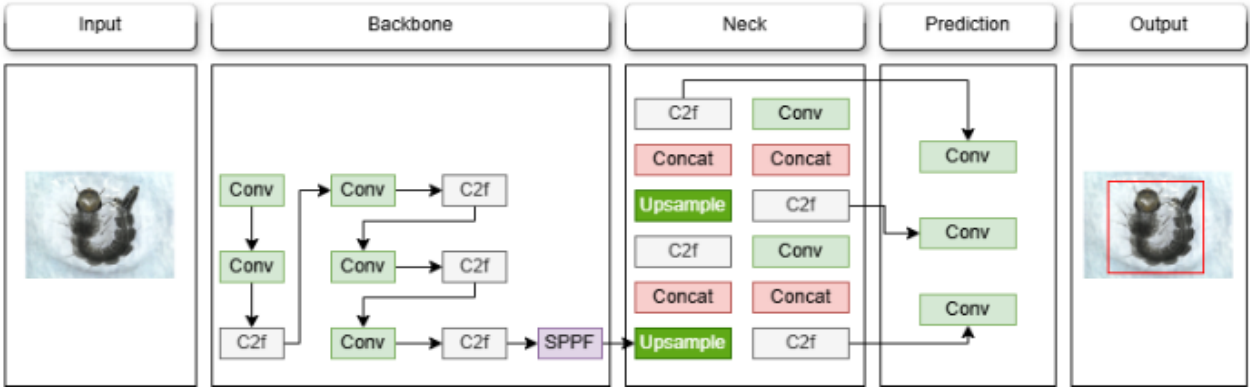


Figure 3: YOLOv8 architecture

We include below a reproducible search procedure and diagnostic criteria to enable reconstruction of a comparable subset.

transparency and replicability of the method while respecting third-party rights. Users can reconstruct a comparable subset by following the search procedure and

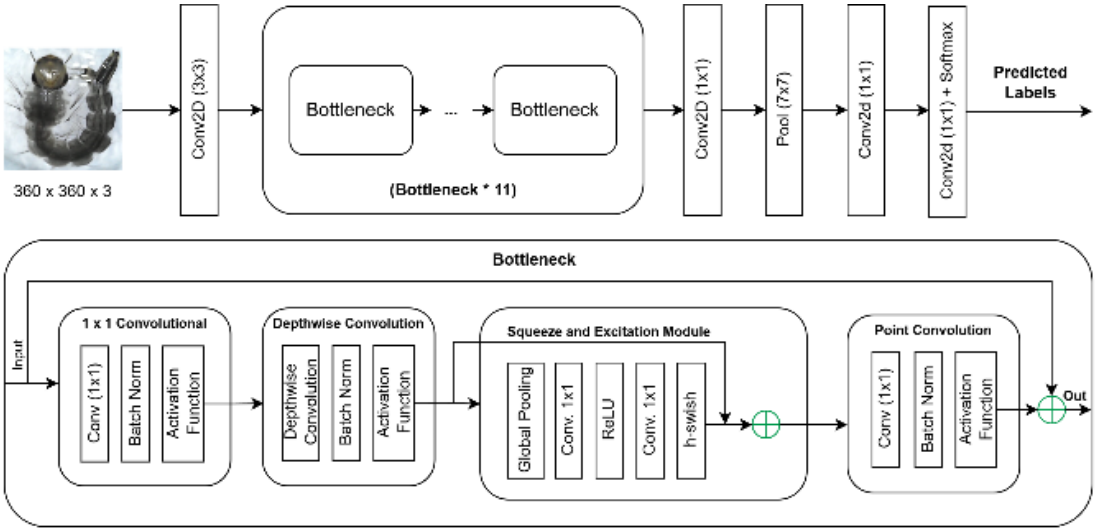


Figure 4: MobileNetV3 architecture

We queried public web sources using combinations of the following terms: “Toxorhynchites larva”, “elephant mosquito larva”, “predatory mosquito larva”, “Toxorhynchites rutilus larva”, and “Toxorhynchites amboinensis larva”. Candidate images/frames were retained only when diagnostic larval characters (below) were visible. Representative background literature describing Toxorhynchites larval biology and morphology is cited to anchor this process.

Morphological verification. Each Unknown image was screened to genus level prior to inclusion.

- Anopheles: required absence of a respiratory siphon and the characteristic horizontal posture parallel to the water surface to be discernible in the image, following identification keys/notes.
- Toxorhynchites: required predatory-larva morphology, including large body size and modified lateral palatal brushes forming an anteriorly directed “basket” during prey capture, as described in entomological literature and inventories; comb and pecten absent.

Ambiguous, low-quality, or non-diagnostic images were excluded.

Harmonization to control domain shift. To limit any cross-source domain shift introduced by mixing institutional and public-web imagery, we applied the same resizing and augmentation policy across Aedes, Culex, and Unknown images (see 3.2 Data Cleaning & Preprocessing).

Data availability and licensing note. Anopheles materials are accessible via Roboflow Universe [27]. For Toxorhynchites, due to heterogeneous licensing and the absence of stable per-image URLs in the early curation, we do not redistribute files and cannot provide a complete URL list. Instead, we document how images were sourced (search recipe above) and how genus identity was verified (diagnostic criteria with citations). This preserves verification criteria cited here.

3.2 Data cleaning & preprocessing

Data preprocessing was conducted to ensure optimal data quality before model training. This process included data augmentation and dataset partitioning. The applied augmentation techniques involved converting a portion of images to grayscale, horizontal and vertical flipping, random rotations (90°, 180°, 270°), hue adjustments (ranging from -10° to +10° degrees), saturation adjustments (-10% to +10%), brightness variations (0–5%), exposure modifications (-5% to +5%), and the addition of noise up to 0.42%. To minimize cross-source domain shift from using different repositories (institutional vs public-web sources), we applied an identical resizing and augmentation policy across Aedes, Culex, and Unknown images.

For the classification task, images detected in the previous stage were cropped based on their bounding boxes and resized to 360×360 pixels. Meanwhile, images for the detection task were resized to 640×640 pixels and annotated using Roboflow. Following preprocessing, the dataset was divided into three subsets: 70% for training, 15% for validation, and 15% for testing.

3.3 YOLOv8-Nano

YOLO (You Only Look Once) was introduced by Redmon et al. (2016) as a novel object detection approach that integrates speed and accuracy into a single prediction stage. Over time, YOLO has undergone several enhancements, one of which is YOLOv8, released by Ultralytics in 2023 [23].

YOLOv8 offers multiple model size variants to accommodate application needs, including YOLOv8-Nano. YOLOv8-Nano is the smallest and lightest variant within the YOLOv8 family, explicitly designed for real-

time detection tasks on resource-constrained devices such as mobile phones or edge computing platforms.

YOLOv8-Nano retains the core architectural structure of YOLOv8, comprising a backbone, neck, and head, but uses fewer parameters to reduce model size and accelerate inference speed. Despite its compact design, YOLOv8-Nano supports multi-scale predictions and adopts an anchor-free approach, which enhances the detection process's simplicity and speed.

In this study, YOLOv8-Nano was selected for its favorable balance of high inference speed, compact model size (approximately 3.2 million parameters), and low resource consumption. These characteristics make it particularly suitable for implementing mosquito larva detection on mobile devices, where real-time performance and computational efficiency are essential.

Figure 3 illustrates the YOLOv8-Nano architecture, highlighting its key real-time detection components: the backbone for feature extraction, the neck for multi-scale feature fusion, and the head for bounding box prediction and classification.

3.4 MobileNetV3 small

Google first developed MobileNet in 2017 as an efficient convolutional neural network architecture designed for mobile and embedded systems [24]. MobileNet utilizes depthwise separable convolution, which separates the filtering and feature combination processes. This significantly reduces the number of parameters and accelerates inference time [25].

MobileNetV3, proposed by [26], is an improvement over its predecessors, introducing several key innovations such as hard-swish activation, squeeze-and-excitation blocks, and a hierarchical block design. MobileNetV3 employs a novel non-linear activation function known as hard-swish (h-swish), a computationally efficient modification of the swish function to reduce computational load and energy consumption while enhancing overall performance. The hard-swish activation function is defined as shown in Equation 1 and is used to minimize the number of training parameters and reduce model complexity and size.

$$\begin{aligned} \mathrm{h-swish}(x) &= x \cdot \alpha(x) \\ \alpha(x) &= \frac{\mathrm{ReLU6}(x+3)}{6} \end{aligned} \tag{1}$$

Where α(x) represents the piecewise linear hard analog function.

MobileNetV3 is available in two variants: Large and Small. In this study, MobileNetV3 Small was chosen due to its lightweight nature and optimization for resource-constrained devices. MobileNetV3 Small combines high computational efficiency with competitive accuracy, making it well-suited for mosquito larva classification tasks on mobile devices.

The MobileNetV3 architecture, depicted in Figure 4, serves as the basis for the classification stage. This process involves feature extraction through convolutional and bottleneck layers, which are key in recognizing visual patterns in larval images. The final output of this process is a class label prediction, such as Aedes, Culex, or Unknown.

3.5 Evaluation measures

Model evaluation is a critical phase in developing object detection and classification systems, including in this study, which employs YOLOv8 for mosquito larvae detection and MobileNetV3 Small for mosquito larvae classification. Evaluation is conducted to assess the model's performance in accurately and efficiently recognizing objects. The evaluation metrics used in this study are described as follows:

Accuracy is employed to evaluate the performance of the mosquito larvae classification model using MobileNetV3 Small. This metric measures the proportion of correct predictions (both positive and negative) relative to the total number of predictions made.

$$accuracy = \frac{(\mathrm{TP} + \mathrm{TN})}{(\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN})} \tag{2}$$

Precision measures the proportion of true positive predictions out of all positive predictions generated by the model. Precision is used in evaluating both the classification model (MobileNetV3 Small) and the object detection model (YOLOv8).

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3}$$

Recall assesses the model's ability to identify all actual positive instances in the data. Similar to precision, recall is utilized in the evaluation of both classification and detection models.

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{4}$$

F1-Score is the harmonic mean of precision and recall and is used to evaluate the balance between the two. This metric is applied in assessing the performance of both classification and detection tasks.

$$\mathrm{F1-Score} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{5}$$

In addition, for the mosquito larvae detection model implemented using YOLOv8, an evaluation metric known as Mean Average Precision (mAP) is employed. This metric is essential for measuring the model's overall performance in object detection, as it considers both the accuracy of bounding box predictions and object classification. The mAP value is derived from the average of the Average Precision (AP) scores across all classes, providing a comprehensive overview of the model's detection accuracy.

$$\mathrm{mAP} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{AP_i} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{6}$$

4 Experimental results

This section presents and analyzes the experimental results obtained from the implementation of mosquito larvae detection and classification methods using the YOLOv8 and MobileNetV3 Small models. This experiment aims to evaluate the effectiveness of both

models in detecting and classifying mosquito larvae based on image data, targeting three specific classes: Aedes, Culex, and Unknown.

4.1 Detection stage with YOLOv8

4.1.1 Model training and hyperparameter tuning

At this stage, the YOLOv8 model was employed to detect mosquito larvae in images with a resolution of 640×640 pixels. The training process involved tuning several key hyperparameters, including the initial learning rate (lr0), the final learning rate (lrf), and the number of epochs. The results of this process are presented in Table 2. The model was evaluated using precision, recall, mAP@0.5, and mAP@0.5:0.95 as performance metrics.

Table 2: Hyperparameter Tuning Results of YOLOv8

No.	lr0	Epoch	mAP @0.5	mAP@ 0.5:0.95	P	R
1.	0.0005	50	98.6	77.7	95.8	96.0
2.	0.00001	30	44.2	30.1	38.1	69.9
3.	0.00001	60	49.2	35.4	39.5	60.9
4.	0.00001	100	75.1	55.8	67.1	69.3

Based on the hyperparameter tuning results, the configuration with a learning rate (lr0) of 0.0005 and 50 epochs provided the best performance due to the balanced convergence speed and stability, as indicated by achieving the highest mAP50 (98.6%) and mAP50-95 (77.7%). Lower learning rates required significantly longer training and yielded lower performance.

4.1.2 Annotation distribution analysis

To ensure the quality of annotation distribution within the dataset, an analysis of bounding box distribution was conducted using a label collegram visualization, as shown in Figure 5. This graph illustrates the distribution of annotation parameter pairs: x, y, width, and height. The visualization results indicate that most objects are located near the center of the image (x ≈ 0.5, y ≈ 0.5), with a relatively wide variation in size. This suggests that the dataset provides good coverage regarding object position and dimensions and is not biased toward specific locations.

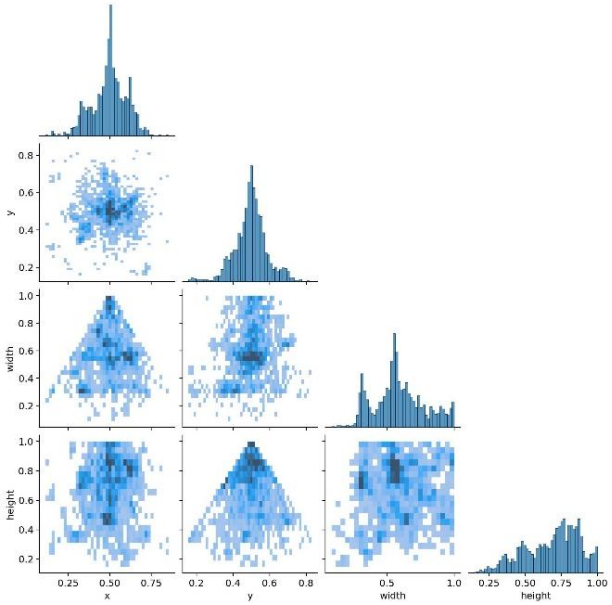


Figure 5: Label Correlogram illustrating annotation distributions within the dataset, highlighting the concentration and variations in object positioning and dimensions. This visualization indicates good coverage and reduces the risk of positional bias

4.1.3 Model training performance evaluation

Figure 6 illustrates the stable downward trend in various loss metrics—such as box loss, classification loss, and distribution focal loss (DFL loss)—indicating that the model has become increasingly proficient in recognizing patterns and minimizing detection errors during training. In addition, evaluation metrics such as precision, recall, and mean Average Precision (mAP) exhibit consistent upward trends, particularly in mAP50 and mAP50-95, which approach optimal values. These results suggest that the YOLOv8 model has been effectively trained and

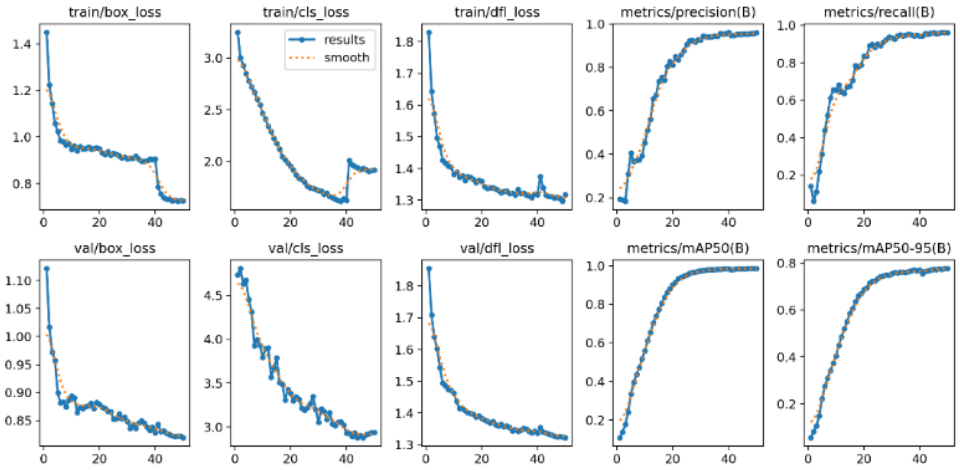


Figure 6: Detection model performance graph

demonstrates strong performance in detecting and classifying mosquito larvae.

4.1.4 Detection evaluation using confusion matrix

To evaluate the classification performance of the mosquito larvae detection model, a confusion matrix was employed, as illustrated in Figure 7. This matrix depicts the distribution of the model's predictions relative to the actual labels across four classes: Aedes, Culex, Unknown, and Background. The main diagonal values represent the number of correct predictions, whereas the off-diagonal values indicate misclassifications between classes.

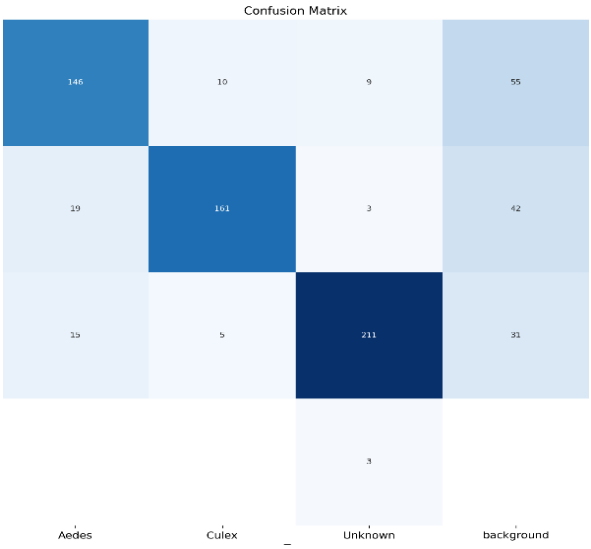


Figure 7: Confusion matrix for the detection stage (YOLOv8-Nano) on the test set. Rows = True class; columns = Predicted class; classes: Aedes, Culex, Unknown, Background

The model performed reasonably well in recognizing larvae from the Aedes class, accurately classifying 146 instances. However, misclassifications occurred: 10 instances were predicted as Culex, 9 as Unknown, and 55 as Background. This suggests a degree of ambiguity between the object and the background in specific samples.

For the Culex class, the model successfully identified 161 instances. Misclassifications included 19 instances labeled Aedes, 3 as Unknown, and 42 as Background. Despite the relatively high accuracy, the misclassification into the background remains a challenge that requires further optimization.

The best performance was observed in the Unknown class, with 211 instances correctly classified. The misclassified instances were relatively low: 15 as Aedes, 5 as Culex, and 31 as Background. This indicates that the model possesses a strong discriminative capability toward the Unknown class, which may include larvae with atypical morphological features.

Meanwhile, a few false positives were observed in the Background class, where 3 instances were incorrectly classified as Unknown. This suggests that although the model can generally distinguish between objects and background, further efforts are needed to mitigate detection errors that may lead to over-detection in real-world implementations, particularly in systems with limited computational resources such as mobile devices.

The confusion matrix indicates that the model can perform object classification satisfactorily, particularly for classes with strong data representation. Nonetheless, background filtering and inter-class balancing improvements are still warranted to further enhance the model's robustness.

Background misclassification analysis. The relatively high number of predictions assigned to the Background class (e.g., 55 for Aedes, 42 for Culex, and 31 for Unknown) suggests several recurring failure modes. First, larvae often appear in low-contrast scenes (e.g., parallel to the water surface) where edges fade into the substrate; second, visual clutter such as detritus, air bubbles, specular reflections, and container boundaries can mimic larval contours; third, small or partially occluded larvae

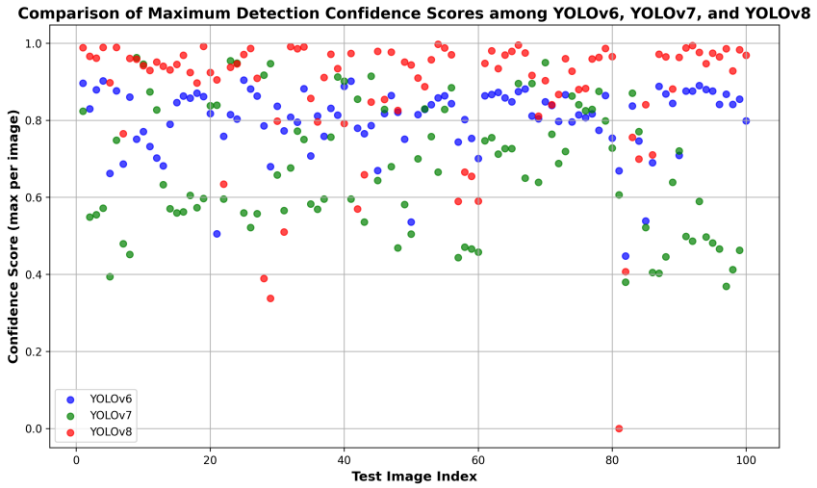


Figure 8: Scatter plot comparing maximum detection confidence scores per test image among YOLOv6, YOLOv7, and YOLOv8 models. The x-axis represents test image samples, while the y-axis indicates detection confidence scores, demonstrating YOLOv8's superior consistency and robustness

may occupy few pixels after resizing to 640×640; and fourth, mild blur caused by water motion reduces fine texture cues. To address these issues, future work will (i) enrich training with hard negatives and background-rich crops, (ii) emphasize small-object sensitivity via larger input resolution or tiling and adopt contrast-focused augmentation (e.g., CLAHE/gamma) in place of heavy de-colorization, (iii) refine ambiguous labels through targeted re-annotation/quality control, and (iv) evaluate a light segmentation/refinement head to better separate larvae from complex water backgrounds. Collectively, these steps are expected to reduce false "Background" assignments while preserving precision.

4.1.5 Analysis of detection confidence scores

Figure 8 presents a scatter plot illustrating the maximum detection confidence score obtained from each test image using YOLOv6, YOLOv7, and YOLOv8. Each point in the plot represents the highest confidence assigned by the respective YOLO model to detected objects within a single test image.

The visualization clearly demonstrates that YOLOv8 achieves the highest and most consistent confidence scores, with the majority of points positioned close to the maximum confidence value of 1.0. This indicates a robust and highly reliable detection capability across the entire test dataset. In comparison, YOLOv7 exhibits moderate variability in confidence scores, typically ranging between approximately 0.5 and 0.9, reflecting somewhat less stable detections. On the other hand, YOLOv6, while competitive, displays lower consistency and several notable outliers below a confidence score of 0.7, indicating lower detection reliability for specific cases.

This comparative analysis emphasizes the superior robustness and detection certainty provided by YOLOv8. Such high and stable confidence levels are crucial for minimizing ambiguous detections and false positives, particularly critical for real-time mobile deployment in practical vector surveillance systems. Therefore, based on these results, YOLOv8 is selected as the primary object detection model for deployment in this study.

4.1.6 Comparison of YOLO versions for optimization

To determine the most optimal detection model for this study, a comparative analysis was conducted between several versions of YOLO, namely YOLOv6, YOLOv7, and YOLOv8. The comparison included evaluations of accuracy (mAP@0.5), training duration, and model size. This evaluation assessed each model's effectiveness and efficiency for mobile deployment. The comparative results are presented in Table 3.

Table 3: Comparison YOLOv6, YOLOv7, and YOLOv8

No.	Model	Map@0.5	Training Time	Model Size
1.	YOLOv6	0.985	1.5 hours	16.9 MB
2.	YOLOv7	0.984	1.8 hours	139 MB
3.	YOLOv8	0.986	0.9 hours	10.7 MB

Based on Table 3, YOLOv8 demonstrated the best performance with an mAP@0.5 score of 0.986, the fastest training time of 0.9 hours, and the smallest model size of 10.7 MB. Although the differences in accuracy among the three models were relatively minor, YOLOv8's efficiency in training time and model size make it the most suitable choice for a mobile-based mosquito larvae detection system. The compact model size facilitates easier integration into devices with limited memory and computational resources.

4.2 Classification stage with MobileNetV3-small

4.2.1 Larvae classification process

After the detection stage is performed using the YOLOv8 model, each detected larvae is marked with a bounding box on the original image. The area enclosed by this bounding box is then cropped to separate the larvae from the background. The cropped result is then used as input for the classification process.

The MobileNetV3-Small model is employed in the classification stage to classify mosquito larvae species into three classes: Aedes, Culex, and Unknown. MobileNetV3-Small was selected based on its lightweight architectural efficiency and optimal operational capability on mobile devices with limited resources.

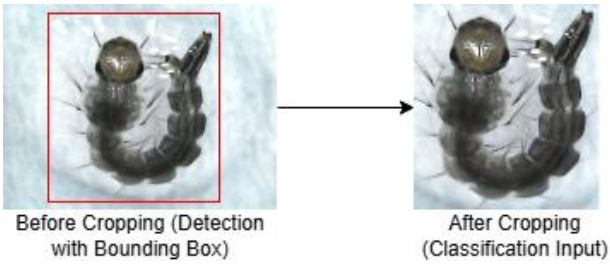


Figure 9: Illustration of the mosquito larvae detection and classification input process. The left image shows the detection result with a bounding box using YOLOv8, and the right image shows the cropped region used as input for classification with MobileNetV3-Small

Figure 9 illustrates the flow of the detection results, where the bounding boxes are applied, and then cropped to create the larvae images, ready to be used as input for the classification stage.

4.2.2 Tuning hyperparameter

To optimize classification performance, tuning was performed on several key hyperparameters: the learning rate and the number of epochs. Evaluation was carried out using accuracy (Accuracy), precision (P), recall (R), and F1-score metrics. The results of the experiments are presented in Table 4.

Table 3: Hyperparameter tuning results of MobileNetV3

No.	lr	Epoch	Accuracy	P	R	F1-Score
1.	0.0001	30	96.2	99.5	99.0	99.2
2.	0.00001	30	69.1	91.4	91.9	91.6
3.	0.00001	60	80.9	98.0	96.6	97.3

Based on the results shown in Table 4, the combination of a learning rate of 0.0001 and 30 epochs was selected as optimal because it delivered superior accuracy (96.2%) and maintained a balanced trade-off between training speed and classification precision compared to lower learning rates and increased epochs, which showed no significant performance improvement.

4.2.3 Classification evaluation using confusion matrix

An analysis was conducted using the confusion matrix, as shown in Figure 10 to evaluate the classification model's performance more comprehensively. This matrix represents the distribution of the model's predictions against the actual labels for the three target classes: Aedes, Culex, and Unknown.

The evaluation results indicate that the model performs excellent classification with a low error rate. For the Aedes class, 208 out of 210 samples were correctly classified, while the remaining two samples were mistakenly predicted as Culex. For the Culex class, 209 out of 210 samples were accurately identified, and one sample was classified as Aedes. For the Unknown class, 208 out of 210 samples were correctly identified, with two misclassifications into the Culex class.

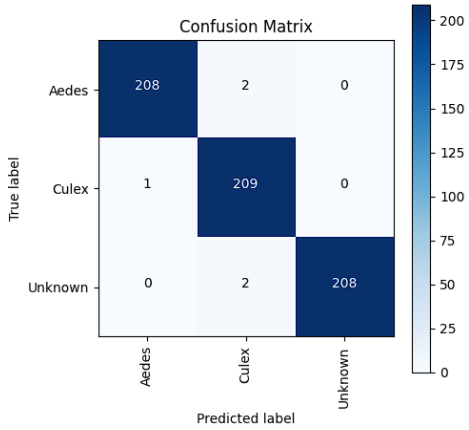
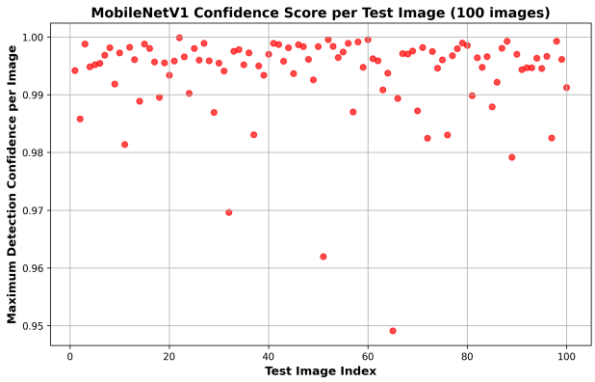
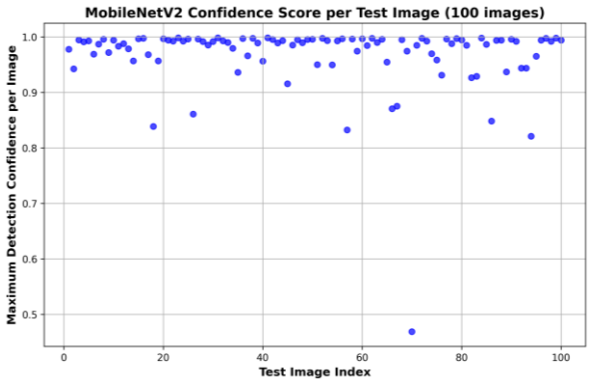


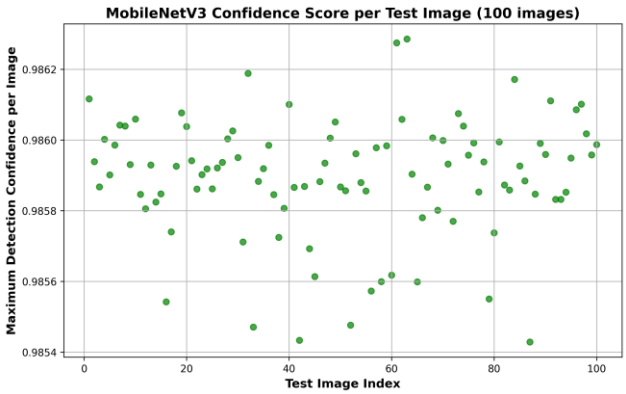
Figure 1: Confusion matrix for the classification stage (MobileNetV3-Small) on the test set. Rows = True class; columns = Predicted class; classes: Aedes, Culex, Unknown



a) MobileNetV1 Confidence Score



b) MobileNetV2 Confidence Score



c) MobileNetV3 Confidence Score

Figure 11: Scatter plot depicting the distribution of classification confidence scores for MobileNetV1, MobileNetV2, and MobileNetV3-Small across test images. MobileNetV3-Small shows consistently higher and more stable confidence scores, supporting its selection for efficient and reliable mosquito larvae classification on mobile devices

4.2.4 Analysis of classification confidence

The scatter plot in Figure 11 shows the distribution of confidence scores generated by the three MobileNet variants—MobileNetV1, MobileNetV2, and MobileNetV3-Small—on each test image. Each point on the scatter plot represents the confidence score of a single test image, providing insights into the stability, consistency, and certainty of the models' classification predictions for mosquito larvae.

From the visualization, it is evident that MobileNetV3-Small consistently produces higher and more stable confidence scores (mostly above 90%) across the test dataset, with less variance compared to MobileNetV1 and MobileNetV2. This indicates that MobileNetV3-Small is not only superior in terms of parameter efficiency and inference speed but also delivers more stable and reliable predictions. In contrast, MobileNetV1 and MobileNetV2 exhibit a wider spread of confidence scores, with several predictions at lower confidence levels, suggesting a higher degree of uncertainty in their classification decisions.

Having high and stable confidence scores is particularly important for real-world deployment, especially for automated mosquito larvae detection and classification on mobile devices. A model with consistently high confidence reduces the risk of false positives and false negatives caused by uncertain predictions. These findings reinforce the results presented in Table 5, where MobileNetV3-Small achieves competitive accuracy with significantly fewer parameters.

Overall, this scatter plot analysis supports the selection of MobileNetV3-Small as the primary classification model in this study, highlighting its advantages not only in efficiency and accuracy but also in providing trustworthy predictions for each test image.

4.2.5 Comparison of MobileNet architectures

In addition to hyperparameter tuning, a comparative study was conducted on three variants of the MobileNet architecture: MobileNet, MobileNetV2, and MobileNetV3-Small. The evaluation focused on four key aspects: validation loss, accuracy on the validation set, final accuracy on the test data, and the total number of parameters for each model. The comparison results are presented in Table 5.

Table 4: Comparison of MobileNet, MobileNetV2, MobileNetV3

Model	Validation Loss	Acc. on Validation Set	Final Accuracy	Total Parameter
MobileNet	0.0283	0.9984	0.9676	3,4m
MobileNet V2	0.0326	0.9984	0.9618	2,5m
MobileNet V3-Small	0.0369	0.9984	0.9625	1m

Although the MobileNet model achieved the highest accuracy (0.9676), it used a larger number of parameters (3.4 million). On the other hand, MobileNetV3-Small demonstrated competitive performance with a final accuracy of 0.9625, while using only 1 million parameters. This efficiency makes MobileNetV3-Small the most suitable choice for deployment on mobile devices with limited memory and computational power. Therefore, in this study, MobileNetV3-Small was chosen as the primary architecture for the mosquito larvae classification stage.

4.3 Detection result

Figure 12 illustrates the detection results of mosquito larvae using an object detection-based system. The image demonstrates that the model can detect two mosquito larva species, Aedes and Culex, by displaying red bounding boxes surrounding the detected larvae, along with class labels and corresponding confidence scores for each prediction. In the top-left image, the larva is identified as Aedes with a confidence level of 75.27%, whereas in the top-right image, the larva is classified as Culex with a confidence of 77.70%. Furthermore, the bottom-left image shows an Aedes larva with a higher confidence score of 87.29%, and the bottom-right image detects a Culex larva

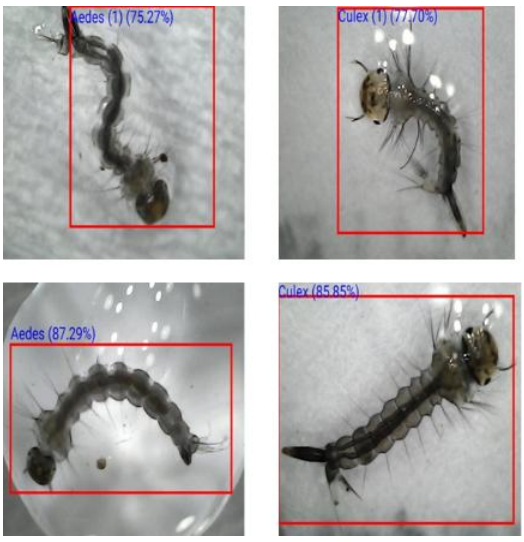
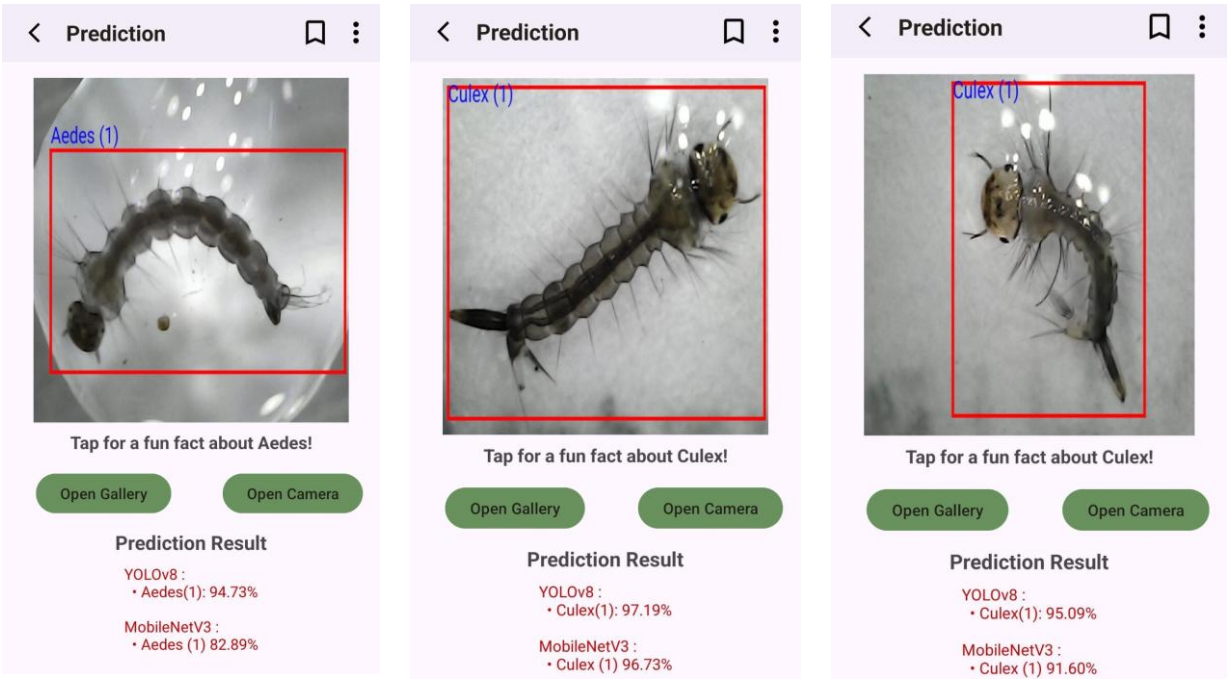


Figure 12: Visualization of mosquito larvae detection results for Aedes and Culex on test images, annotated with bounding boxes and confidence scores.



a) Aedes (YOLOv8 94.73%; MobileNetV3 82.89%) b) Culex (YOLOv8 97.19%; MobileNetV3 96.73%) c) Culex (YOLOv8 95.09%; MobileNetV3 91.60%)

Figure 13: On-device detection (YOLOv8-Nano) and classification (MobileNetV3-Small) running in our mobile app. The app displays the detector label and a "Prediction Result" panel listing both YOLOv8 and MobileNetV3 outputs.

with a confidence of 85.85%. These relatively high confidence values across all detections indicate the model performs well in distinguishing between mosquito larva species based on microscopic images. These results suggest that the developed system can be employed as an effective tool for the automatic and efficient identification of mosquito larvae, which is highly beneficial for vector-borne disease surveillance and control efforts.

4.4 On-device deployment and real-time results

To validate real-time feasibility on mobile hardware, we deployed the cascaded YOLOv8-Nano → MobileNetV3-Small pipeline in an Android app. The app performs on-device detection and then classifies each detected larva; both outputs are rendered on screen. Figure 13 presents three representative screenshots with aligned detector–classifier predictions: panels (a) and (b) show Culex (YOLOv8: 95.09% / 97.19%; MobileNetV3: 91.60% / 96.73%), while panel (c) shows Aedes (YOLOv8: 94.73%; MobileNetV3: 82.89%).

5 Discussion

The experimental results demonstrate that the combination of YOLOv8 and MobileNetV3-Small architectures can detect and classify mosquito larvae with high accuracy while maintaining efficiency in mobile deployment. Achieving a mAP@0.5 of 98.6% in the detection stage and classification accuracy of 96.2% indicates that this integrated system excels in precision and inference speed.

These findings are consistent with previous studies, such as [15], where YOLOv5 and FPSNet were employed to detect and classify three types of mosquito larvae with high accuracy. However, the study did not explicitly address model efficiency in mobile environments and maintained a separate two-stage computational process for detection and classification, which entails higher resource demands. In contrast, this study strategically selected the lightweight and efficient MobileNetV3-Small architecture to overcome limitations while integrating detection and classification into a unified and mutually reinforcing pipeline.

Furthermore, compared to studies [21] and [22], which focused solely on larvae detection (primarily Aedes) without specific species classification, our approach adds value by directly identifying Aedes and Culex at the genus level while assigning non-Aedes/Culex specimens—such as Anopheles, Toxorhynchites—to an "Unknown" class. Thus, the present work provides an integrated and real-time two-stage pipeline (detection + 3-class classification: Aedes, Culex, Unknown). Distinguishing Anopheles as a dedicated class is left for future work.

Additionally, selecting MobileNetV3-Small as the classification model demonstrates that a lightweight architecture can still achieve competitive accuracy. As shown in Table 5, although the original MobileNet achieved a slightly higher final accuracy, its significantly more significant number of parameters (3.4 million) makes it less ideal for mobile implementation than

MobileNetV3-Small, which has only 1 million parameters. These findings support the arguments made in studies [15] and [8], which emphasize the importance of model efficiency for field applications. However, their approaches remain limited by the number of classes classified or by focusing on binary classification (larvae vs. non-larvae).

These results have significant implications, particularly in vector-borne disease control in endemic areas. Genus-level identification of Aedes and Culex—combined with an "Unknown" bucket that signals the presence of other genera (e.g., Anopheles, Toxorhynchites)—can assist health workers or the public in taking more targeted preventive actions, such as fogging or eliminating mosquito breeding grounds, by the primary disease vectors. As outlined in the Conclusion, adding a dedicated Anopheles class is planned as future work.

In conclusion, this study offers a technically superior approach and makes a practical contribution by supporting an adaptive, mobile-friendly vector surveillance system that is well-suited for field deployment under resource-constrained conditions.

6 Conclusion

Based on the analysis, design, implementation, and testing processes, this study successfully developed a mosquito larvae detection and classification system using YOLOv8-Nano and MobileNetV3-Small. The system demonstrated high performance, achieving a classification accuracy of 96.2% and a mean Average Precision at Intersection over Union of 0.5 (mAP50) of 98.6%. MobileNetV3-Small's efficiency enables real-time deployment on commodity mobile devices. The system can identify mosquito larvae from the Aedes and Culex genera, and recognize an 'Unknown' class that, in this study, explicitly comprises the genera Anopheles and Toxorhynchites.

This research's theoretical contribution lies in integrating the anchor-free YOLOv8 detection architecture with the lightweight MobileNetV3 classification model into a unified system optimized for mobile platforms. This approach provides a novel foundation for developing multi-class object detection systems in environmental biology domains where high efficiency and precision accuracy are essential. Furthermore, this study enhances our understanding of how modern convolutional networks can utilize the spatial representations of mosquito larvae to enable automatic genus-level classification.

From a practical perspective, the proposed system offers an innovative solution for supporting environmental health management, particularly in efforts related to vector surveillance and control, with increased speed and accuracy. By enabling automatic and real-time larvae detection, the system can potentially reduce reliance on time- and resource-intensive manual identification. Its field deployment could facilitate faster responses to potential outbreaks and improve the effectiveness of mosquito control programs.

For future development, it is recommended to expand the diversity and quantity of training data to enhance the model's generalization capabilities across varying lighting conditions, backgrounds, and larval types. The explicit inclusion of the Anopheles genus as a distinct class could further broaden the taxonomic scope of classification. Employing newer versions of YOLO may also contribute to improved efficiency and accuracy through lighter architectures and advanced optimization techniques. Interdisciplinary collaboration with entomology experts is strongly advised to ensure taxonomic validity and maximize the system's practical utility. Lastly, integrating geolocation features could extend the system's benefits toward mapping mosquito larvae hotspots and supporting more comprehensive spatially driven vector control strategies.

References

[1] S. Prasher and L. Nelson, "Mosquitoes Classification using EfficientNetB4 Transfer Learning Model," 2023, ICAAIC, pp. 582–586, 2023, doi: 10.1109/ICAAIC56838.2023.10141504.

[2] World Health Organization, "Vector-borne diseases," World Health Organization. Accessed: Feb. 16, 2025. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/vector-borne-diseases

[3] G. Liang, X. Gao, and E. A. Gould, "Factors responsible for the emergence of arboviruses; strategies, challenges and limitations for their control," Emerging Microbes and Infections, vol. 4, no. 3, 2015, doi: 10.1038/emi.2015.18.

[4] M. R. Marselle, J. Stadler, H. Korn, K. N. Irvine, and A. Bonn, "Vector-Borne Diseases," in Biodiversity and Health in the Face of Climate Change, Springer, 2019, pp. 67–90. doi: 10.1007/978-3-030-02318-8.

[5] M. Parra-Amaya, M. Puerta-Yepes, D. Lizarralde-Bejarano, and S. Arboleda-Sánchez, "Early Detection for Dengue Using Local Indicator of Spatial Association (LISA) Analysis," Diseases, vol. 4, no. 2, p. 16, 2016, doi: 10.3390/diseases4020016.

[6] E. S. Paixão, M. G. Teixeira, and L. C. Rodrigues, "Zika, chikungunya and dengue: The causes and threats of new and reemerging arboviral diseases," BMJ Global Health, vol. 3, pp. 1–6, 2018, doi: 10.1136/bmjgh-2017-000530.

[7] J. V. J. Silva, T. R. R. Lopes, E. F. d. Oliveira-Filho, R. A. S. Oliveira, R. Durães-Carvalho, and L. H. V. G. Gil, "Current status, challenges and perspectives in the development of vaccines against yellow fever, dengue, Zika and chikungunya viruses," Acta Tropica, vol. 182, no. December 2017, pp. 257–263, 2018, doi: 10.1016/j.actatropica.2018.03.009.

[8] M. S. Saeed, S. F. Nazreen, S. S. S. A. Ullah, Z. F. Rinku, and A. Rahman, “Detection of Mosquito Larvae Using Convolutional Neural Network,” International Conference on Robotics, Electrical and Signal Processing Techniques, pp. 478–482, 2021, doi: 10.1109/ICREST51555.2021.9331235.

[9] W. D. M. De Silva and S. Jayalal, “Dengue mosquito larvae identification using digital images,” Proceedings - International Research Conference on Smart Computing and Systems Engineering, SCSE 2020, pp. 31–36, 2020, doi: 10.1109/SCSE49731.2020.9313003.

[10] S. I. Park et al., “Species identification of food contaminating beetles by recognizing patterns in microscopic images of elytra fragments,” PLoS ONE, vol. 11, no. 6, pp. 1–22, 2016, doi: 10.1371/journal.pone.0157940.

[11] J. Wang, C. Lin, L. Ji, and A. Liang, “A new automatic identification system of insect images at the order level,” Knowledge-Based Systems, vol. 33, pp. 102–110, 2012, doi: 10.1016/j.knosys.2012.03.014.

[12] H. P. Yang, C. Sen Ma, H. Wen, Q. Bin Zhan, and X. L. Wang, “A tool for developing an automatic insect identification system based on wing outlines,” Scientific Reports, vol. 5, pp. 1–11, 2015, doi: 10.1038/srep12786.

[13] R. Charoenpanyakul et al., “Enhancing mosquito classification through self-supervised learning,” Scientific Reports, vol. 14, no. 1, pp. 1–20, 2024, doi: 10.1038/s41598-024-78260-2.

[14] P. I. Nainggolan et al., "Classification Of Aedes Mosquito Larva Using Convolutional Neural Networks And Extreme Learning Machine," 2023 7th International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM), Medan, Indonesia, 2023, pp. 79-83, doi: 10.1109/ELTICOM61905.2023.10443125.

[15] Pauzi Ibrahim Nainggolan, Syahril Efendi, Mohammad Andri Budiman, Maya Silvi Lydia, Romi Fadillah Rahmat, Dhani Syahputra Bukit, Umi Salmah, Sri Malem Indirawati, and Riza Sulaiman, "Detection and Classification of Mosquito Larvae Based on Deep Learning Approach," Engineering Letters, vol. 33, no. 1, pp198-206, 2025.

[16] M. Valan, K. Makonyi, A. Maki, D. Vondráček, and F. Ronquist, “Automated Taxonomic Identification of Insects with Expert-Level Accuracy Using Effective Feature Transfer from Convolutional Networks,” Systematic Biology, vol. 68, no. 6, pp. 876–895, 2019, doi: 10.1093/sysbio/syz014.

[17] I. A. Rather et al., “Prevention and control strategies to counter dengue virus infection,” Frontiers in Cellular and Infection Microbiology, vol. 7, no. JUL, pp. 1–8, 2017, doi: 10.3389/fcimb.2017.00336.

[18] J. A. Luna et al., “Label-free visualization of internal organs and assessment of anatomical differences among adult Anopheles, Aedes, and Culex mosquito specimens using bidirectional optical coherence tomography,” Optics and Laser Technology, vol. 168, no. July 2023, p. 109849, 2024, doi: 10.1016/j.optlastec.2023.109849.

[19] N. D. G. Drantantiyas et al., “Performasi Deteksi Jumlah Manusia Menggunakan YOLOv8,” JASIEK (Jurnal Aplikasi Sains, Informasi, Elektronika dan Komputer), vol. 5, no. 2, pp. 63–68, 2023, doi: 10.26905/jasiek.v5i2.11605.

[20] A. F. Félix-Jiménez, V. S. Sánchez-Lee, H. A. Acuña-Cid, I. Ibarra-Belmonte, E. Arredondo-Morales, and E. Ahumada-Tello, “Integration of YOLOv8 Small and MobileNet V3 Large for Efficient Bird Detection and Classification on Mobile Devices,” AI (Switzerland), vol. 6, no. 3, pp. 1–27, 2025, doi: 10.3390/ai6030057.

[21] A. M. Hubalde, D. A. Padilla, and D. A. C. Santos, “A yolo-based approach for Aedes aegypti larvae classification and detection,” 2021. ICIVC 2021, pp. 161–167, 2021, doi: 10.1109/ICIVC52351.2021.9527017.

[22] R. Abedin, A. Al Nakib, T. Islam, and M. E. Islam, “An Improved Transfer Learning Based Larvae Detection and Classification Using Densenet121,” 2023. ICCIT, pp. 1–6, 2023, doi: 10.1109/ICCIT60459.2023.10441505.

[23] B. Gašparović, G. Mauša, J. Rukavina, and J. Lerga, “Evaluating YOLOV5, YOLOV6, YOLOV7, and YOLOV8 in Underwater Environment: Is There Real Improvement?”. 2023, SpliTech 2023, pp. 12–15, 2023, doi: 10.23919/SpliTech58164.2023.10193505.

[24] H. Abdo, K. M. Amin, and A. M. Hamad, “Fall Detection Based on RetinaNet and MobileNet Convolutional Neural Networks,” Proceedings of ICCES 2020, doi: 10.1109/ICCES51560.2020.9334570.

[25] A. G. Howard et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” 2017, doi: 10.48550/arXiv.1704.04861.

[26] A. Howard et al., “Searching for mobileNetV3,” Proceedings of the IEEE International Conference on Computer Vision, vol. 2019-Octob, pp. 1314–1324, 2019, doi: 10.1109/ICCV.2019.00140.

[27] Roboflow Universe, “Mosquito Larvae Dataset (Thesis),” Dataset webpage, 2022–2025. Available: https://universe.roboflow.com/thesis-elw7x/mosquito-larvae-dataset-zfjs9. Accessed: Mar. 2, 2