VOL. 20, NO. 2, JANUARY 2025 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

LAMOKollect: A CONVOLUTIONAL NEURAL NETWORK (CNN) CLASSIFICATION MODEL FOR MOSQUITOES OF MEDICAL IMPORTANCE

Jessica S. Velasco[1], Jhon Mark S. Blay[1], Alethea Coleen R. Rañosa[1], Benjie U. Monte De Ramos[1], Dave Carlos A. Calixto[1] and Benedicto N. Fortaleza[2,1]
[1]Department of Electronics Engineering, College of Engineering, Technological University of the Philippines, Manila, Philippines
[2]University Extension Services, Technological University of the Philippines, Manila, Philippines
E-Mail: jessica_velasco@tup.edu.ph

ABSTRACT

Malaria is one of the most life-threatening diseases transmitted to people through the bites of female mosquitoes of the genus Anopheles that were infected with protozoan parasites belonging to the genus Plasmodium. Anopheles mosquitoes are considered intermediate hosts or vectors of malaria in humans, which to this day still lacks preventive measures in controlling the growth of vector mosquitoes and identification of their species. This problem is being addressed in this study considering that mosquitoes from the genus Anopheles are of medical importance and could be used as a model in classifying certain species of Anopheles mosquitoes that are vectors of the malaria disease in humans through a novel species identification model. Convolutional Neural Networks (CNNs) are designed for image and pattern recognition, using hierarchical layers of convolutions to automatically learn and extract relevant features from input data. The input data is an image database composed of 957 images from six Anopheles species, namely Anopheles coustani, Anopheles crucians, Anopheles funestus, Anopheles gambiae, Anopheles punctipennis, and Anopheles quadrimaculatus, which have significant physical damage in many specimens, reflecting real-world conditions. To address the fine-grained morphological diversity and high intra- and inter-species variation, seven CNN models were employed and evaluated using transfer learning. Performance was assessed based on accuracy, loading time, and model weight size. GoogleNet achieved the highest accuracy of 95%, while YOLOv8 was the fastest and most lightweight model in terms of loading time and weight size, respectively. These findings demonstrate the potential of deep learning for mosquito species classification even with challenging image data, paving the way for automated vector identification in public health applications.

Keywords: convolutional neural networks, anopheles, vector mosquitoes, malaria, mosquito classification, deep learning.

Manuscript Received 15 November 2024; Revised 25 January 2025; Published 10 March 2025

1. INTRODUCTION

Mosquitoes are among the deadliest animals on the planet; they carry a wide range of humanity's most fatal diseases, including malaria, dengue fever, Zika, West Nile virus, Chikungunya virus, and yellow fever that infect between 200-500 million people every year. Over 40% of the world's population is at risk from these mosquito-borne illnesses, which are becoming a major worldwide health concern (Franklinos *et al*., 2019). Merely 90 species, or less than 3% of the 3,500 officially identified species of mosquitoes, are known to transmit disease. Furthermore, it is believed that less than 10% of mosquitoes can spread disease, and this is mostly composed of three genera, namely Aedes, Anopheles, and Culex out of 112 mosquito genera. However, no efficient medications or vaccines for diseases such as dengue and Zika are made available to the public, and vector control is still heavily dependent on precise identification of these disease vectors to implement targeted interventions, which can lower transmission rate and inhibit the growth of mosquito population (Namias *et al*., 2021). Furthermore, traditional insecticide-based strategies, including chemicals, remained as the major approach to controlling mosquito populations, but insecticide resistance and drug-resistant pathogens have widely emerged in mosquitoes in recent years, deliberately causing more death due to mosquito-borne diseases.

Malaria, a mosquito-borne disease spread among humans by a parasite that commonly infects a particular kind of mosquito, is a life-threatening disease characterized by high fever, severe chills, and sweating. According to a World Health Organization (WHO), it is estimated that malaria caused 247 million illnesses and 619,000 deaths in 2021, and existing malaria control efforts have not seen any discernible improvement (WHO, 2023a). Malaria is curable and preventable if diagnosed and treated promptly; it does not spread from person to person. However, if left untreated, a patient may develop severe complications or even death. This particular disease is transmitted to people through the bites of a certain group of infected female mosquitoes from the genus Anopheles.

Anopheles mosquitoes are considered intermediate hosts or vectors of malaria in humans, which are caused by these mosquitoes carrying any of the five parasitic species of Plasmodium, and two of these species - *Plasmodium vivax* and *Plasmodium falciparum* - pose the greatest threat (WHO, 2023). There are about 460 species of Anopheles, but only 30-40 commonly transmit the malaria parasite Plasmodium. Anopheles mosquitoes

ARPN Journal of Engineering and Applied Sciences

also carry filariasis, encephalitis, heartworms, and O'nyong nyong fever. This problem is being addressed in this study considering that mosquitoes from the genus Anopheles are of medical importance and could be used as a model in classifying certain species of Anopheles mosquitoes that are vectors of the malaria disease in humans through a novel species identification model.

Various mosquito identification models have already been developed and are proven to be effective in terms of identifying mosquitoes of medical importance or those who are found to be intermediate hosts of mosquito-borne parasitic diseases. Accurate identification of the mosquito vectors is crucial as various species exhibit diverse biological characteristics (e.g. host preference and biting patterns) which affect vector competence and inform effective strategic initiative (Erlank *et al*., 2018). Typically, mosquito recognition requires the microscopic and biochemical analysis of hundreds of mosquitoes per week and the use of dichotomous keys in identification of species (Rueda, 2004), which is a time-consuming activity that requires complex scaling. Fortunately, several automated methods have been developed in mosquito species identification, including wingbeat frequencies analysis, computer vision, and most notably the Convolutional Neural Network (CNN) in recognizing mosquito images.

Computer vision, particularly machine learning (ML), has demonstrated remarkable efficacy in the past few decades in addressing various public health concerns, ranging from diagnosis, prognosis, and in treating illnesses in people to forecast outbreaks of infectious diseases. It is rapidly being developed for application in public health and medicine, where it has resolved problems that other computational methods had encountered and produced appropriate interpretations in several biological research subfields, including pharmacogenomics. CNN has proven to be particularly efficient in reading "Bigdata" and identifying potential diseases with a high level of accuracy (Jaiswal *et al*., 2019). Computer vision also holds a potential as a quantifiable method for accurate and reliable identification of mosquitoes, such that, like mosquito taxonomists, it relies on visual (morphological) features to assign a definitive taxon name (Goodwin *et al.*, 2021). With the potential applications of CNNs in identifying different types of mosquitoes, which are vectors of many infectious diseases, such as dengue, chikungunya, and Zika, health workers and entomologists can quickly and accurately classify mosquitoes from easy-to-acquire images, without the need for specialized training or equipment. This can help in the prevention and control of mosquito-borne diseases, as well as in the study of mosquito ecology and behavior.

Various studies from the past have used CNNs to classify mosquitoes, such as that of Matto *et al.* (2019), which achieved an accuracy of 83.9% for distinguishing between *Aedes aegypti*, *Culex quinquefasciatus*, and *Aedes albopictus*, three of the most common and important vector mosquitoes. Kumar *et al.* (2022) achieved an impressive 99.2% accuracy with CNN amongst species of Aedes and Culex mosquitoes.

In this study, images of wild-caught mosquitoes, specifically from the genus Anopheles, were acquired from a dataset available online. The majority of the wild-caught specimens in the image database created for this system's training and testing bear substantial physical damage, obscuring some diagnostic characteristics. With that problem and given the fine-grained morphological diversity between closely related Anopheles species, as well as the high inter- and intraspecific diversity, the experiment will be simply defined as a noisy, fine-grained, open set recognition object classification problem. The findings of this study could be useful for public health professionals in identifying mosquito vectors in the surrounding area.

2. METHODOLOGY

A. Datasets

In the study, wild-caught mosquito species identified through morphology or by DNA barcoding were used.

This image database is made available online by Goodwin, *et al.* (2021) under a Creative Commons Attribution 4.0 International License from their study about developing an ensemble model for novel mosquito species detection that can be accessed through https://github.com/vectech-dev/novel-species-detection. In this database, 12,977 photos of 2,696 specimens from 67 species of mosquitoes can be found.

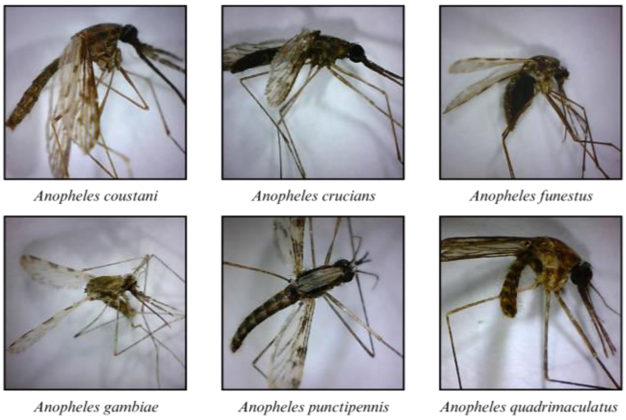

Figure-1. Example image dataset samples of wild-caught Anopheles mosquitoes (Goodwin *et al*., 2021).

However, in this study, only 6 species from the genus Anopheles were utilized to focus on identifying vectors of the malaria disease, namely *Anopheles coustani, Anopheles crucians, Anopheles funestus, Anopheles gambiae, Anopheles punctipennis, and Anopheles quadrimaculatus* species (see Figure-1). The dataset was divided into two batches, for training and testing/validation. There are a total of 957 images in the dataset, consisting of 668 images for Train data, 192 for

Validation, and 97 for Test data; all images are in a .jpg file extension stretched to 640×640 pixels. The number of images per class is summarized in the supplementary table below.

Table-1. Summary of the sample size of the dataset used.

Anopheles species	Number of Images	Train Data	Validation Data	Test Data
A. coustani	163	102	42	19
A. crucians	163	115	31	17
A. funestus	163	119	28	16
A. gambiae	163	116	27	20
A. punctipennis	163	114	35	14
A. quadrimaculatus	142	102	29	11
Total	957	668	192	97

The dataset is divided into three batches, namely the Train, Valid, and Test data. As shown in Table-1, the dataset consists of 957 images, broken into 70%, 20%, and 10% for the Train, Valid, and Test set, respectively.

B. Machine Learning Platforms, Libraries, and Frameworks

a. Google Colab

Google Colab, short for Colaboratory, is a free, cloud-based platform provided by Google that facilitates the development and execution of deep learning models, such as Convolutional Neural Networks (CNNs). It offers a Jupyter notebook environment with access to powerful TPU (Tensor Processing Unit) and GPU (Graphics Processing Unit) resources, enabling users to train and speedrun CNNs without the need for high-end hardware. Google Colab is also compatible with popular libraries such as TensorFlow, Keras, PyTorch, and OpenCV (Plagata, 2020).

b. Keras

Keras is a high-level neural networks API in Python, serving as an interface for creating, training, evaluating, and deploying deep learning models. It provides a user-friendly and modular framework, allowing users to easily design neural network architectures by stacking layers. Keras supports various backends, such as TensorFlow or Theano, and offers built-in functions for common deep learning operations. Additionally, Keras offers the necessary libraries for loading the dataset and dividing it into training sets (X_train), which is used for fine-tuning of the net, and testing sets (X_test), which is used for performance assessment. (Gulli & Pal, 2017).

c. Pytorch

PyTorch is an open-source machine learning library that is primarily used for building and training deep learning models. It is developed by Facebook's AI Research lab (FAIR) which provides a dynamic computational graph, allowing for more intuitive model development and easier debugging. These tensors encode the model's parameters, including its inputs and outputs. For fast prototyping models, PyTorch offers a variety of handy building blocks for defining neural networks. (Rickson, J. 2022).

d. Roboflow

Roboflow is a computer vision developer framework that helps with data collection, preprocessing, and model training techniques. Roboflow provides implementations of popular object detection models, such as YOLOv3 and YOLOv5, in both Keras and PyTorch frameworks (Siddique *et al.*, 2023). It has public datasets readily available to users and allows users to upload their custom data. This custom data can be used in training models and be exported for inference. In this experiment, Google Colab was used as the programming platform for Keras and PyTorch with a TensorFlow backend. Specifically, Keras platform was used in training DenseNet (201 and 121), ResNet-101, YOLOv8n and Xception models, while PyTorch was utilized in training GoogleNet model. This was done this way to analyze how different frameworks result in varying model performance. The image dataset used in this experiment was uploaded in the Roboflow database and was trained using Roboflow 2.0 classification model. It was then used in training the models listed above for validation accuracy. A 224×224×3 input image was used for all models except Xception, which used a 229×229×3 image size. Further, varying batch size, such as 32, 128, and 256, while training was performed for 20 epochs each. Batch size is the number of training instances observed before the optimizer performs a weight update while epochs is the number of times the model is exposed to the training set (Gulli & Pal, 2017). Lowering the batch size was done to allow the GPU to run faster and more efficiently. Loss and accuracy were recorded for each model to show how these two vary in both the training and validation set for each epoch. Confusion matrix was then generated to calculate various performance metrics, such as accuracy, recall, precision, and F1 score.

C. Training the CNN model architectures

Through Convolutional Neural Networks, the baseline accuracy expected from modern deep learning models can be established when classifying vector mosquitoes with high inter-species similarity and intra-species variations. Specifically, investigating the effectiveness of transferring features acquired from a generic dataset into the classification of vector mosquitoes requires the application of several different fine-tuning techniques or architectures. In this experiment, seven CNN

VOL. 20, NO. 2, JANUARY 2025 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

models were adapted, namely DenseNet-121 and -201, GoogleNet (InceptionV1), MobileNetV2, ResNet-50V2 and -101, YOLOv8n and Xception. The training was performed only on the approximately 70% training dataset and the remaining 30% dataset was reserved for validation (20%) and testing (10%). The dataset is mounted from Google Drive to Colab to read the path linked to the Train, Valid, and Test folders.

a. DenseNet-121 and -201

Densely Connected Convolutional Networks (DenseNet) is a feed-forward convolutional neural network (CNN) architecture that uses the combined knowledge of all the previous layers in the network architecture to link each layer to every other layer (Huang *et al.*, 2016).

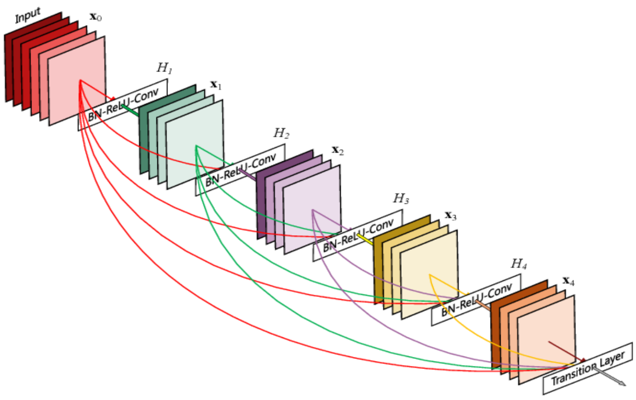

Figure-2. Architecture of DenseNet. (Huang *et al.*, 2016).

Dense models use layer stacking as an alternative to adding multiple layers. Since each layer comprises the knowledge from all the previous layers, the network is more compact and has fewer channels. The layers are also narrow, which makes it more efficient and smaller in size. DenseNet-121 and DenseNet-201 specifically denote different variants with 121 and 201 layers, respectively. These models excel in image classification tasks, feature reuse, and parameter efficiency by leveraging dense connectivity, leading to improved accuracy and reduced training time compared to traditional architectures like ResNet. DenseNet-121 is a lighter version suitable for resource-constrained applications, while DenseNet-201 provides a deeper and potentially more expressive model for tasks requiring higher capacity and complexity. For training the 121- and 201-layer deep DenseNet model on the data, a batch size of 128 and 20 epochs were used. A 224×224 pixel crop was used and the image shape was rescaled into [0, 1] range.

b. GoogleNet & InceptionV3

Google Net (or Inception V1) was proposed by researchers at Google (with the collaboration of various universities) in 2014 in the research paper titled “Going Deeper with Convolutions”. Compared to earlier cutting-edge architectures like AlexNet and ZF-Net, the GoogLeNet architecture is significantly different because it uses many different kinds of methods such as 1×1 convolution and global average pooling that allow it to create a deeper architecture.

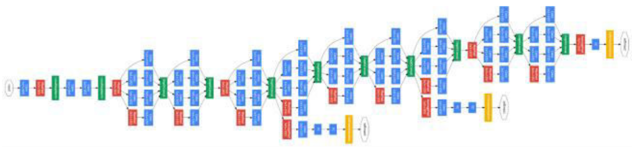

Figure-3. GoogleNet Model Architecture.

The overall architecture is 22 layers deep and it was designed to keep computational efficiency at ease. The reason being is that the architecture can be run on individual devices even with low computational resources. The architecture also contains two auxiliary classifier layers connected to the output of Inception (4a) and Inception (4d) layers. This architecture captures images of size 224 x 224 with RGB color channels. All the convolutions within this architecture use Rectified Linear Units (ReLU) as their activation functions.

On the other hand, InceptionV3, which also belongs to the Inception family, makes several improvements including using label smoothing, factorized 7×7 convolutions, and the use of an auxiliary classifier to propagate label information lower down the network. The InceptionV3 is a superior version of the basic model Inception V1 (GoogleNet) and it aims to be computationally efficient and highly accurate on image classification tasks.

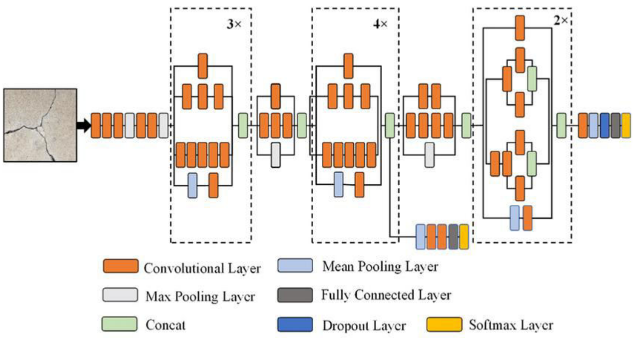

Figure-4. InceptionV3 Model Architecture (Ali *et al.*, 2021).

The Inception V3 model consists of 42 layers, organized into several inception modules. Each module has parallel branches with different types of convolutions and pooling operations. In this experiment, the learning rate was set to 0.001, decaying every 20 epochs at a rate of 0.13 for both GoogleNet and InceptionV3 models.

c. ResNet

The Residual Network (ResNet) is a model with an extremely deep CNN composed of 152 layers that utilize residual blocks (Figure-4) to maintain the identity function of previous parts. With the use of this

ARPN Journal of Engineering and Applied Sciences

architecture, sequence models that maintain the context of earlier levels can be created. Two 3x3 layers make up each residual block, which are then followed by a normalization layer and an activation function called a Rectified Linear Unit (ReLU) (He *et al.*, 2015) which results in the outputs with the same shape as the inputs. ResNet models have been widely used for applications such as image recognition and classification, due to their resemblance to VGG models.

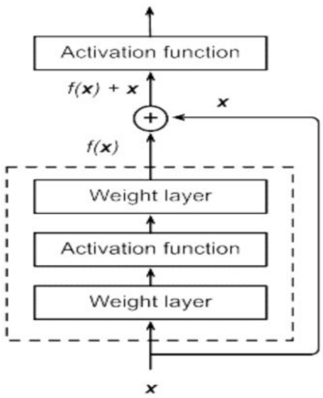

Figure-5. Schematic representation of a residual block. (He *et al.*, 2016).

ResNet-101 specifically, is a model composed of 101 layers. This architecture excels in image classification tasks, leveraging residual connections to mitigate the vanishing gradient problem and improve the flow of information through the network. ResNet-101 provides additional capacity for tasks requiring higher expressive power. This model is faster and more accurate than some previous CNN models, such as VGG-16 and VGG-19. ResNet-101 is a 3-layer deep model that was trained on the data with a batch size of 32, 20 epochs, and randomized learning rate. The image is resized with its shorter side randomly sampled from range [0,255] to range [0, 1] with a randomly sampled 224×224 crop.

d. Xception

Xception, which stands for "Extreme Inception", is a deep learning architecture created by Google. The Xception architecture has 36 convolutional layers that serve as the network's foundation for feature extraction. These convolutional layers are structured into 14 modules, all of which have linear residual connections around them, with the first and last modules as exceptions (Chollet, 2017).

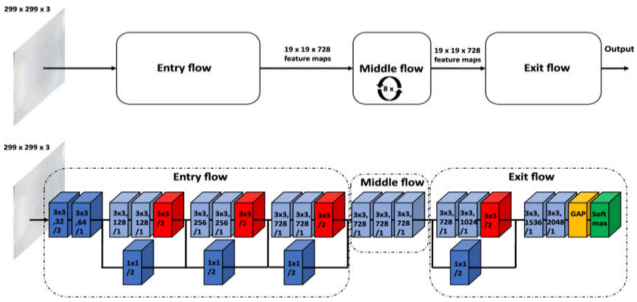

Figure-6. Xception CNN architecture (Westphal *et al.*, 2021).

A linear stack of depth-wise separable convolution layers with residual connections makes up the Xception architecture (Figure-6) which makes it very easy to define and modify; it takes only 30 to 40 lines of code using a high level library such as Keras or TensorFlow, not unlike an architecture such as VGG-16. Here, the Xception model used a 229×229 input shape and a batch size of 128, which were trained from 20 epochs. The pixel values are resized to [0, 1] range to normalize the input value of an image.

e. YOLOv8

YOLO (You Only Look Once) is a popular set of object detection models used for real-time object detection and classification in computer vision. YOLOv8 is the newest state-of-the-art YOLO model that was launched in January 2023 by its developers, Ultralytics, who also created the influential and industry-defining YOLOv5 model. This model can be used for object detection, image classification, and instance segmentation tasks.

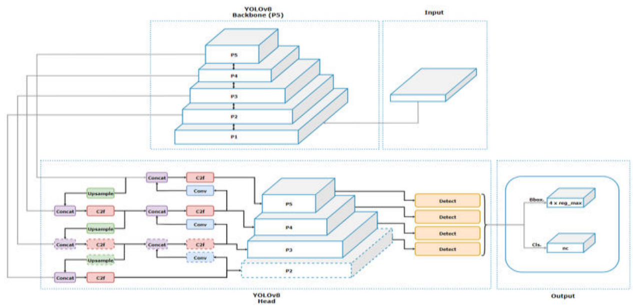

Figure-7. YOLOv8 CNN architecture (Karna *et al.,* 2023).

The improved YOLOv8 network architecture includes an additional module for the head represented in the rectangle with a dashed outline. YOLOv8 includes numerous architectural and developer experience changes and improvements over YOLOv5. The architecture of YOLOv8 builds upon the previous versions of YOLO algorithms. YOLOv8 utilizes a convolutional neural network that can be divided into two main parts: the backbone and the head (Figure-7). Using this model, we analyzed its performance using the imported custom dataset from RoboFlow.

VOL. 20, NO. 2, JANUARY 2025 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

3. RESULTS AND DISCUSSIONS

Based on comparison and evaluation from the experiments conducted, the confusion matrix, loading time, accuracy, and weight size of the pre-trained convolutional neural network are used to assess its effectiveness in classifying mosquito species.

A. Confusion Matrices

Confusion matrix aids with prediction accuracy improvement, misclassification identification, and model performance analysis. The column represents the actual class and the row represents a predicted class. Figure shows the confusion matrices of the 6 mosquito species.

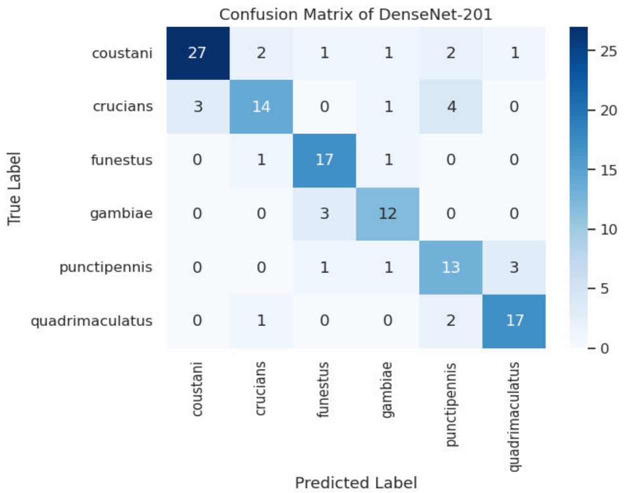

Figure-8. DenseNet-201 Confusion Matrix.

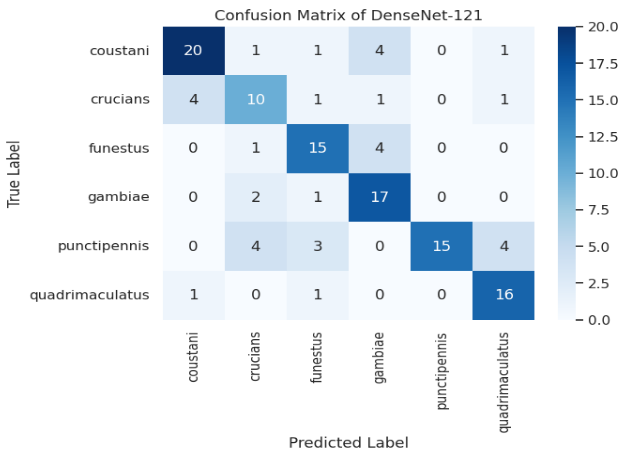

Figure-9. DenseNet-121 Confusion Matrix.

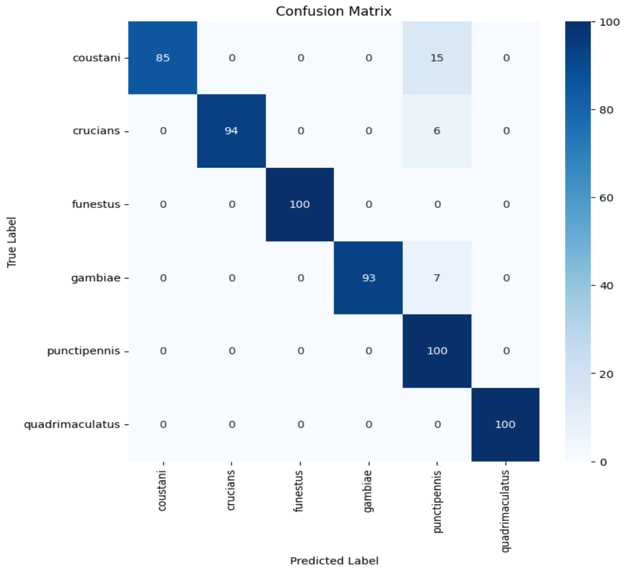

Figure-10. GoogleNet Confusion Matrix.

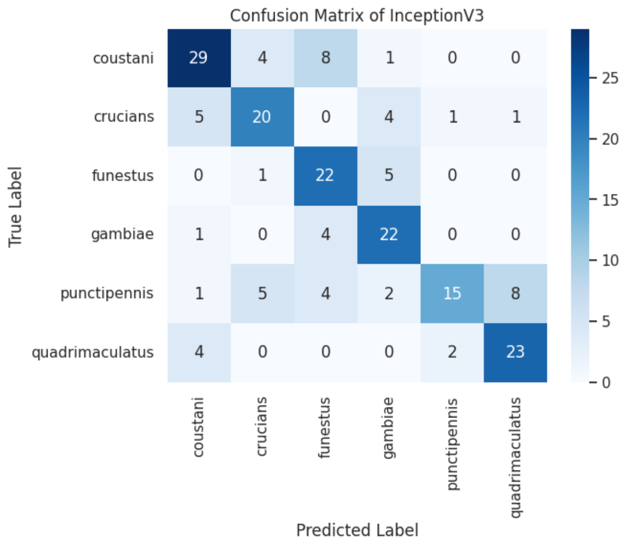

Figure-11. InceptionV3Confusion Matrix.

VOL. 20, NO. 2, JANUARY 2025 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

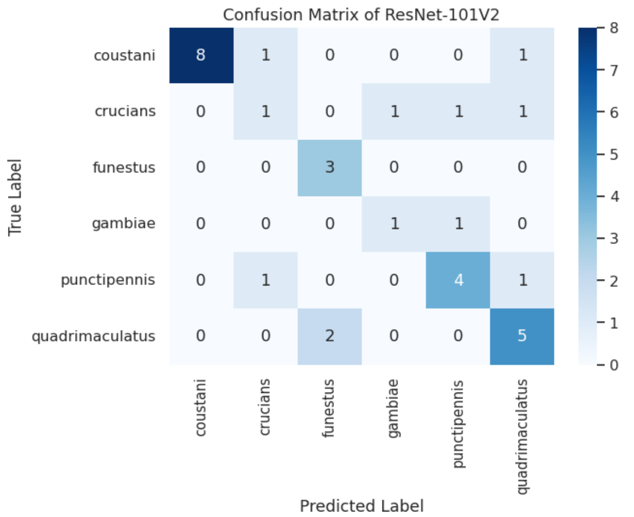

Figure-12. ResNet-101V2 Confusion Matrix.

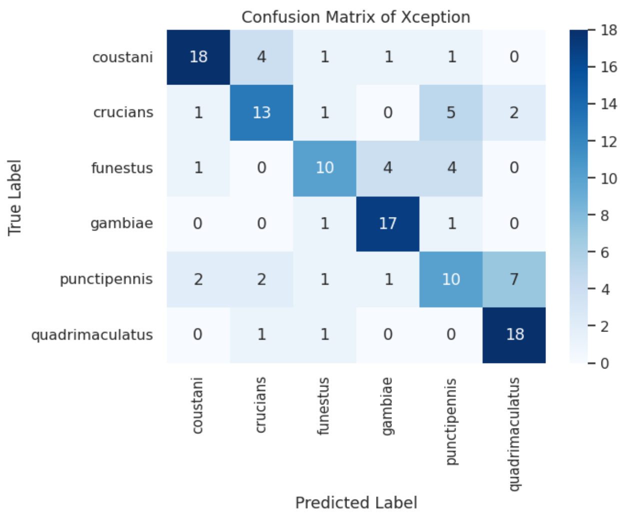

Figure-13. Xception Confusion Matrix.

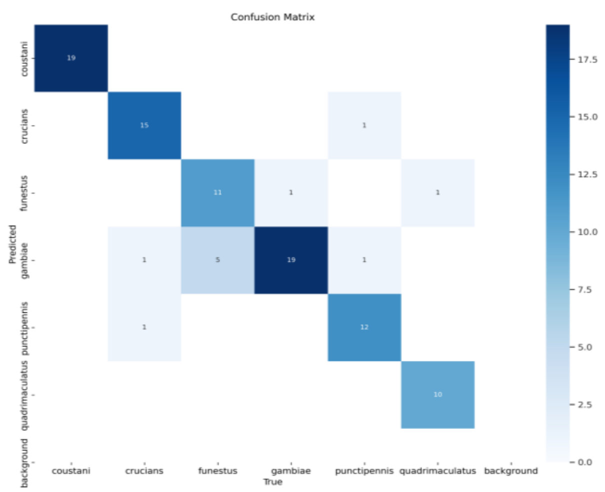

Figure-14. YOLOv8 Confusion Matrix.

B. Weight Size, Loading Time, and Accuracy

Among the seven models, GoogleNet provides the highest accuracy with a 95% accuracy along with its 49.61 MB weight size with a loading time of 2 seconds. In terms of loading time, YOLOV8 was the fastest to load as well as the lightest model with a weight size of 3.20 MB. However, with a 67% accuracy percentage, 84.27 MB weight size, and a loading time of 7 seconds - Xception has the lowest accuracy recorded.

Table-2. Evaluation of each Model by its Weight Size, Loading Time, and Accuracy.

Model	Weight Size (MB)	Loading Time (seconds)	Accuracy (%)
DenseNet-201	438.86	4	78%
DenseNet-121	223.64	3	73%
GoogleNet	49.61	2	95%
InceptionV3	83.7	3	68%
ResNet-101V2	946.66	5	69%
Xception	84.27	7	67%
YOLOv8	3.20	0.001	89%

C. Accuracy and Loss Graph

The performance of the models is displayed on the accuracy and loss graph. Accuracy graph compares the model predictions with the true values in percentage terms to determine how well our model predicts. Whereas the loss graph determines the total errors the model represents. Training and Validation accuracy and loss plotted in the graph will be interpreted to evaluate the performance of the model.

Figures 8-14 show the accuracy and loss graph over the number of epochs for the 6 classifications of mosquito species.

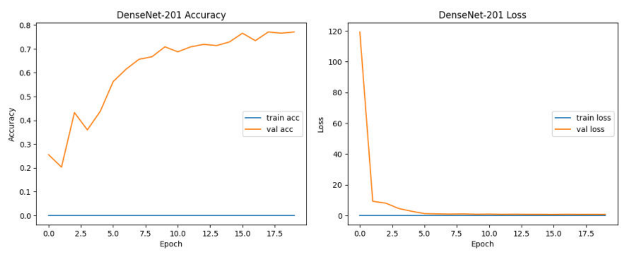

Figure-15. DenseNet-201 Accuracy and Loss Graph.

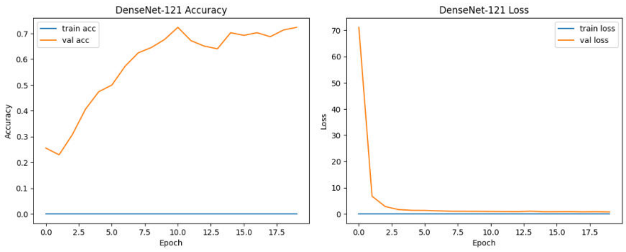

Figure-16. DenseNet-121 Accuracy and Loss Graph.

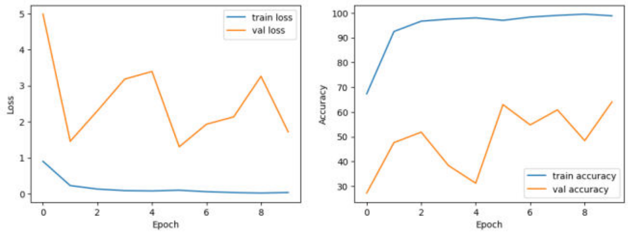

Figure-17. GoogleNet Accuracy and Loss Graph.

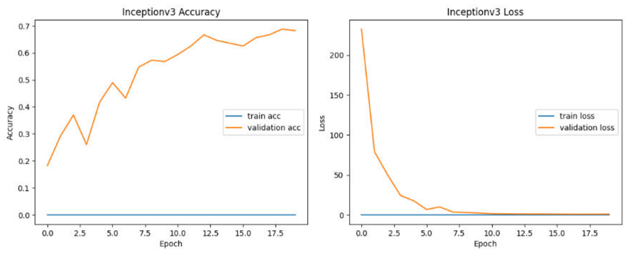

Figure-18. Inception V3 Accuracy and Loss Graph.

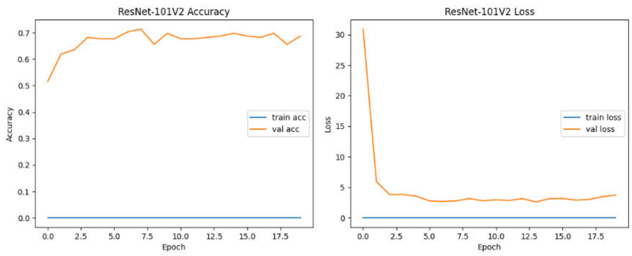

Figure-19. ResNet-101V2 Accuracy and Loss Graph.

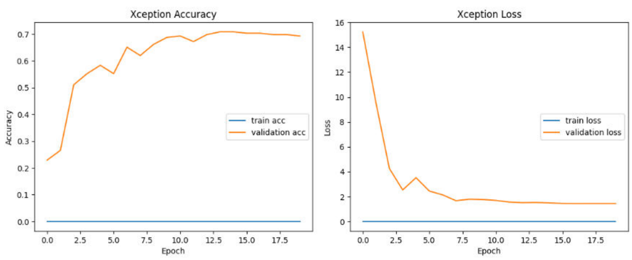

Figure-20. Xception Accuracy and Loss Graph.

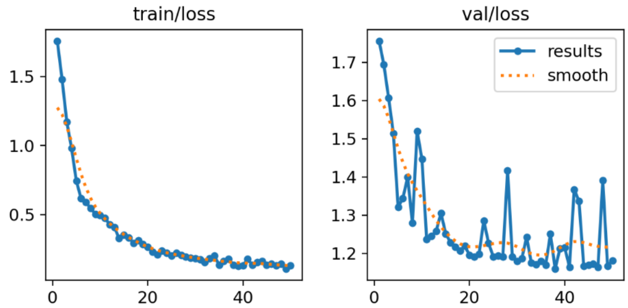

Figure-21. YOLOv8 Accuracy and Loss Graph.

As the graphs suggest, the loss graph of DenseNet-201 and -121, Inception, ResNet-101V2, and Xception shows a decreasing trend, indicating that the error between predicted and actual values is low. This could also mean that these models are performing well on the training data and that the predicted values are close to the true values. On the other hand, accuracy for all models shows an increasing trend, indicating that they have improved performance and are making correct predictions on a large proportion of the input data.

4. CONCLUSIONS

Using deep neural networks and transfer learning, this paper demonstrated an image-based classification of mosquito species. An image database of six species of Anopheles mosquitoes was utilized, containing 957 images, 668 of which are for Train data, 192 for Validation, and 97 for Test data. Performance evaluation was based on confusion matrix, loading time, accuracy, and weight size of the pre-trained convolutional neural network. As a result, GoogleNet is the model with the highest accuracy of 95% with 49.61 MB weight size and loading time of 2 seconds. The fastest to load as well as the lightest model was YOLOV8 with 0.001 second loading time and 3.20MB weight size, respectively. Consequently, these findings can greatly help in identifying vectors of the malaria disease in native regions with high densities of carrier mosquitoes, preventing the spread of malaria in humans.

5. RECOMMENDATION

Future researchers can evaluate the optimal model that can be derived using various datasets when categories in a dataset are large or very small to improve the accuracy level of the models used in this work. It would also be best to try various batch sizes, epochs, and other parameters regarding the chosen architecture. We also recommend exploring a range of models, including popular and new ones, as well as employing ensemble methods by combining predictions from multiple CNN models to produce a more reliable outcome.

ACKNOWLEDGEMENT

The researchers express their sincere thanks to Adam Goodwin and his team for making the dataset and codes that were used in this study available for public use under a Creative Commons Attribution 4.0 International License. The authors also thank Engr. Jessica S. Velasco for her critical remarks and technical assistance during their study. Further, they would also like to credit their friends for their valuable contribution and assistance in the progress of their work.

REFERENCES

Ali L., Alnajjar F., Jassmi H. A., Gocho M., Khan W. and Serhani M. A. 2021. Performance evaluation of deep CNN-based crack detection and localization techniques for concrete structures. *Sensors*. https://doi.org/10.3390/s21051688

Chollet F. 2017. Xception: Deep Learning with Depth Wise Separable Convolutions. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/CVPR.2017.195.

Erlank E., Koekemoer L. and Coetzee M. 2018. The importance of morphological identification of African anopheline mosquitoes (Diptera: Culicidae) for malaria control programmes. Malar. J.

Franklinos L., Jones K., Redding D., Abubakar I. 2019. The effect of global change on mosquito-borne disease. *Lancet Infect. Dis*. https://doi.org/10.1016/s1473-3099(19)30161-6

Goodwin A., Padmanabhan S., Hira S. *et al.* 2021. Mosquito species identification using convolutional neural networks with a multitiered ensemble model for novel species detection. *Sci Rep.* https://doi.org/10.1038/s41598-021-92891-9

Gulli A. and Pal S. 2017. Deep Learning with Keras. *Packt* Publishing. https://doi.org/10.5555/3153803

He K., Zhang X., Ren S. and Sun J. 2015. Deep Residual Learning for Image Recognition. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2016.90

Huang G., Liu Z. and Weinberger K. 2016. Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/CVPR.2017.243

Jaiswal A., Tiwari P., Kumar S., Gupta D., Khanna A., Rodrigues J. 2019. Identifying Pneumonia in Chest X-rays: A Deep Learning Approach. University of Bedfordshire. https://doi.org/10.1016/j.measurement.2019.05.076

Karna N., Putra M. A. P., Rachmawati S. M., Abisado M. and Sampedro G. A. 2023. Towards Accurate Fused Deposition Modeling 3D Printer Fault Detection using Improved YOLOv8 with Hyperparameter Optimization. IEEE Access. https://doi.org/10.1109/ACCESS.2023.3293056

Kumar V., Prasad V., Kunaisa K., Swain M., Anandaram H. and Kumar A. 2022. Mosquito Type Identification using Convolution Neural Network. 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC). https://doi.org/10.1109/ICOSEC54921.2022.9951979

Motta D., Santos A., Winkler I., Machado B., Pereira D., Cavalcanti A., *et al*. 2019 Application of convolutional neural networks for classification of adult mosquitoes in the field. PLoS ONE. https://doi.org/10.1371/journal.pone.0210829

Namias A., Jobe N., Paaijmans K., Huijben S. 2021. The need for practical insecticide-resistance guidelines to effectively inform mosquito-borne disease control programs. Elife. https://doi.org/10.7554/eLife.65655

Plagata T. 2020. Setting Up Google Colab for CNN Modeling. Medium. https://medium.com/swlh/setting-up-google-colab-for-cnn-modeling-55b5208599c4

Rickson J. 2022. What Is PyTorch? Builtin. https://builtin.com/machine-learning/pytorch

Rueda L. 2004. Pictorial keys for the identification of mosquitoes (Diptera: Culicidae) associated with Dengue Virus transmission. Zootaxa.

Siddique S., Islam E., Neon E., Sabbir T., Naheen I. and Khanm R. 2023. Deep Learning-based Bangla Sign Language Detection with an Edge Device. Intelligent Systems with Applications

Westphal E. and Seitz H. 2021. A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks. Additive Manufacturing https://doi.org/10.1016/j.addma.2021.101965

World Health Organization. 2023a. World malaria report 2022. Retrieved from https://www.who.int/publications/i/item/9789240064898

World Health Organization. 2023. Malaria. Retrieved from https://www.who.int/health-topics/malaria#tab=tab_1