Machine Learning Enabled Edge Intelligence for Monitoring Singapore's Aedes Mosquito Population

Isabelle Ong Li Xin[1], So Hoi Ling[2], Tan Peng Loon Alex[3], Lai Tien Leong Andrew[3], Teo Jenn Yue[3]

[1]Raffles Institution (Junior College), 1 Raffles Institution Lane, Singapore 575954
[2]Hwa Chong Institution (College), 661 Bukit Timah Rd, Singapore 269734
[3]DSO National Laboratories, 12 Science Park Drive, Singapore 118225

Abstract

Dengue is a severe disease which is increasingly prevalent in Singapore. Thus, it is imperative to find a solution to better monitor the population of Aedes mosquitoes, which transmit dengue. This research presents key contributions in the two areas: ML optimisation for robustness and edge networking. Past research involving ML identification of mosquito species focuses on ML architecture, with little discussion of practical deployment. This research is a novel contribution as a use-case of an ML approach for Aedes mosquito identification under different practical conditions, and has found that an amalgamation of data augmentation techniques is best for improving model performance. Another gap in current research is the lack of investigation of the application of networking in large-scale monitoring of the Aedes mosquito population. However, this is vital to reduce manpower in Aedes population monitoring by networking the Gravitraps at the edge to a command centre. Therefore, this research addresses this gap by demonstrating a novel integration of the state-of-the-art ML model YOLO (You Only Look Once) with networking on edge devices for remote monitoring of Aedes population with 100% accuracy and real-time networking with inference speed of 112ms on an edge device.

1 Introduction

1.1 Background and Rationale for Choice of Topic

Dengue is a disease transmitted to humans through the bite of an infected Aedes mosquito [1]. About 1 in 20 people who are infected with dengue will develop severe symptoms, which can result in shock, internal bleeding, and even death [2].

According to the National Environmental Agency (NEA), the Aedes mosquito population in the Singapore community was about 48 per cent higher in March this year (2022) compared to the same period in 2021 [3]. This points to a worrying trend of an increasing number of dengue cases in Singapore.

Given the severity of the disease, as well as its increasing prevalence among the populace, it is imperative to find a solution to reduce the spread of dengue. Currently, traps are deployed to monitor the distribution of mosquitoes in Singapore. The existing mechanism deployed by NEA is the Gravitrap [4]– this attracts female Aedes adult mosquitoes looking for sites to lay their eggs with a hay-infused water, before trapping them with a sticky lining [5]. Despite its advantage of its simplicity in design, there are limitations to the effectiveness of this mechanism in reducing dengue cases. The primary drawback is the significant amount of human intervention involved— the Gravitraps are checked on a fortnightly basis to ensure proper functioning and mosquitoes have to be sent to a lab for manual identification [6].

Fig. 1 Image of Gravitrap

1.2 Proposed Solution

In this study, a Machine Learning (ML) approach combined with networking across devices is proposed to remotely and effectively differentiate Aedes mosquitoes from non-Aedes mosquitoes in the Gravitrap to reduce cost and manpower required to track the distribution of Aedes mosquitoes across the country.

Currently, research into using ML to differentiate between various species of mosquitoes primarily focus on the ML architecture rather than the deployment of the model in practical scenarios. For example, Kittichai et al. examined the robustness of their ML model in identifying the species of mosquitoes, but did not go beyond that to look at the practical deployment of the model [7]. Yin et al. also investigated the accuracy of an ML model that classified mosquito species based on their wingbeat sounds, but similarly did not look into the practical deployment of their model in a specific use case [8]. Meanwhile, Surya et al. compared the robustness of different ML networks, but did not do so with reference to a specific practical deployment [9]. As for socket networking, there is currently no available research into its deployment in the identification of mosquitoes in edge devices.

Therefore, this research paper presents the following key contributions:

a) Investigation into the deployment of ML model in a specific use case (ie. in the Gravitraps around Singapore). The deployment of the ML model for mosquito classification in this practical situation poses many challenges, such as potential poor lighting, poor camera resolution and multiple mosquitoes of different species in the trap at the same time. Thus, this research is a novel contribution to this field as it looks into addressing these issues during optimisation of the ML model for robust classifications under practical conditions.
b) Integration of networking with use of ML model at the edge. This is novel because no research has been done to combine ML with networking in this field, and also solves issues that come with the specific use-case of ML at the Gravitraps, such as low power of the edge devices.

2 Methodology

2.1 Data Collection and Processing

The data was collected from various sources to ensure diversity. For Aedes mosquitoes, the images were obtained from Kaggle and additional Internet sources, while the images of non-Aedes mosquitoes were obtained from Dryad and additional Internet sources (see appendix). Such diverse sources were chosen to prevent homogeneity of the dataset, which could negatively impact the performance of the dataset through overfitting [10].

To prevent class imbalance which negatively affects the performance of the ML model [11], the proportion of images of Aedes mosquitoes and non-Aedes mosquitoes were kept at 50-50. The dataset was then split in an optimal 70-20-10 manner across the training, validation and test datasets.

2.2 ML Approach

You Only Look Once (YOLO) is used as the ML model. The YOLO detection network has 24 convolutional layers followed by 2 fully connected layers, and the algorithm works by dividing the image into N grids, each having an equal dimensional region of SxS. Each of these N grids is responsible for the detection and localization of the object it contains. These grids then predict B bounding box (bbox) coordinates relative to their cell coordinates, along with the object label and probability of the object being present in the cell [12]. The detection process consists of three stages– pre-processing, inference and non-max-suppression (NMS). Pre-processing transforms the data into a more understandable format, and addresses potential issues with the data [13]. Meanwhile, inference involves utilising data points to calculate the classification output [14]. Lastly, NMS addresses the issue of multiple bboxes being generated for each image, by selecting the best bbox for an object and "suppressing" all other overlapping bboxes [15].

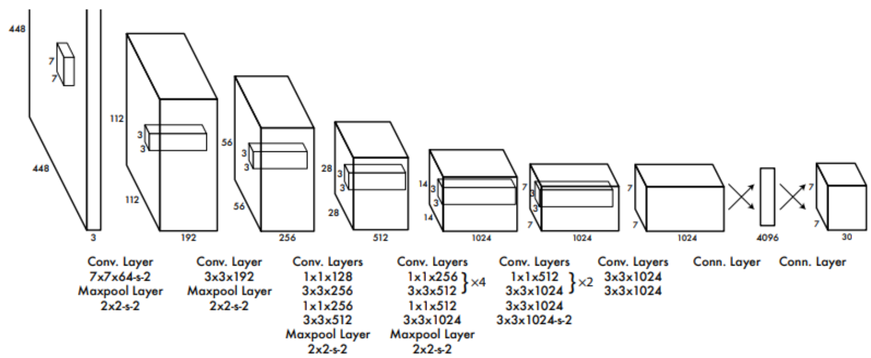


Fig. 2 Overview of YOLO architecture

YOLO is a state-of-the-art object detector, which has a relatively high mean average precision (mAP) and much faster speeds than other real-time object detection networks such as R-CNN and DPM [12]. This is a critical consideration in this research because the detection system should ideally be both fast and accurate to quickly and accurately send information about the classification of mosquitoes in the traps. This allows information on the population distribution of Aedes mosquitoes to be compiled in real-time, successfully achieving the aim of efficient monitoring of the Aedes mosquito population in Singapore.

2.3 Model Optimisation

Optimising the performance of the ML model can be done primarily through improving the dataset and altering the hyperparameters of the code [16]. With each parameter changed, the performance of the model was evaluated through the proportion of true positives as reflected in the confusion matrix, mean average precision (mAP) and the number of successful classifications of the bboxes. The proportion of true positives was the focus in the analysis of the confusion matrix because the main aim of the ML model is to identify Aedes mosquitoes accurately– the accurate identification of non-Aedes mosquitoes is less consequential. The background column in the confusion matrix refers to other background objects missed by the detector, which holds little significance in this research given the black background in the images taken of mosquitoes in the Gravitrap [17].

2.3.1 YOLO Version

First, the effect of the YOLO version used on the performance of the model was investigated. The experimentation dataset size was kept constant at 600 images (300 images per class), and the hyperparameters were kept constant at 8 batches and 50 epochs.

2.3.2 Dataset Size
Next, the effect of dataset size on the performance of the model was investigated. The dataset size was varied between 200 images (100 images per class), 600 images (300 images per class) and 1000 images (500 images per class). The hyperparameters were kept constant at 8 batches and 50 epochs.

2.3.3 Hyperparameter Optimisation
To improve the performance of the model, hyperparameter tuning was investigated, specifically the hyperparameters of batch and epoch number. This was done through random sampling, one of the most commonly used methods for hyperparameter optimisation, where a search space of hyperparameter values was defined before testing values in that domain [18].

2.3.4 Data Augmentation
The effect of data augmentation on the performance of the model was then investigated as this is crucial given that the quality of images captured in the Gravitrap might be compromised by environmental conditions. The optimal model of YOLOv5 and optimal dataset size of 1000 images were used with the hyperparameters kept constant at 8 batches and 50 epochs. The models were compared by having them detect the same set of 100 images that included images augmented with the various augmentation techniques (20 per augmentation technique, and 20 non-augmented images), and comparing their accuracy and ability to generate bboxes around the mosquitoes.

<u>2.3.4.1 Blurring</u>
The first effect carried out to augment the dataset was blurring. This effect was chosen because in the trap, the sharpness of the image cannot be guaranteed due to potential pertinent limitations, such as camera resolution with the small size of the mosquitoes, and the camera shutter speed with the possible motion of the mosquitoes. Out of the 1000 images, 100 were chosen from each class to be blurred (see appendix).

<u>2.3.4.2 Addition of Noise</u>
The next effect carried in data augmentation was the addition of noise. This effect was chosen because the amount of noise in images could be affected by factors such as electricity and heat [19], which is present in the trap when the camera and ML model are deployed. Out of the 1000 images, 100 were chosen from each class to have noise added (see appendix).

<u>2.3.4.3 Merging Images</u>
Merging of images was also conducted (see appendix) due to the consideration that there may be more than one mosquito in the trap at any point of time, which could result in the image capturing more than one mosquito. Therefore, it is essential to ensure that the model can identify the species of mosquitoes even when there is more than one mosquito present.

<u>2.3.4.4 Changing Exposure</u>
Another data augmentation method tested was changing the exposure of images (see appendix), the rationale being that in the trap, the exposure of the images may differ at different times of the day.

2.3.4.5 Amalgamation of Dataset Augmentation Techniques

After investigating different dataset augmentation techniques individually, these techniques were combined to create the optimal dataset. Out of 1000 images, 100 images for each class for each augmentation were included. The goal was to have an optimal dataset which is trained on highly diversified data to be able to accurately classify Aedes mosquitoes in varied environments.

2.4 Networking for Edge Intelligence

As a proof of concept, networking was conducted between Raspberry Pis representing the deployment of the ML model on the islandwide mosquito traps, and a Jetson Nano representing the command centre. Factors such as where image processing was conducted were investigated to optimise the efficiency of the network.

Socket programming was utilised for networking. Sockets facilitate the transmission of information across the network, allowing data to be centralised from edge devices to the main server [20]. The network involves client-server communication, where the server is a powerful computer with high processing power while the client relies on the server for more CPU-intensive processing tasks such as model inference [21].

Two Raspberry Pis were used to represent the deployment of the ML model at the Gravitraps. They are optimal as they are low-power devices that do not need to be connected to a power source and can operate from batteries or a power bank, and more than one was used to show the possibility of having multiple traps around the island connected to a command centre. Meanwhile, the Jetson Nano is a higher-power device that needs to be connected to a power source, hence it was used to represent the command centre.

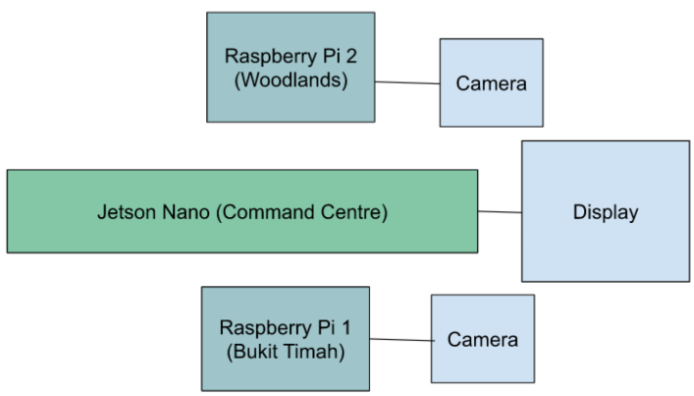


Fig. 3 Overview of hardware involved in networking stage

3 Results and Discussion

3.1 Model Optimisation

The effects of YOLO version, dataset size, hyperparameter optimisation and data augmentation techniques on the performance of the model were investigated.

3.1.1 YOLO Version

When YOLOv3 was used, the proportion of true positives was 0.95, while the proportion of true negatives was 1.00 and the mAP was 0.995. The proportion of test images with bboxes, meanwhile, was 46.7% (28 out of 60 images).

When YOLOv5 was used, the proportion of true positives was 0.98, while the proportion of true negatives was also 0.98. The mAP was 0.994, and the proportion of test images with bboxes was 100% (60 out of 60 images). (See appendix for confusion matrices)

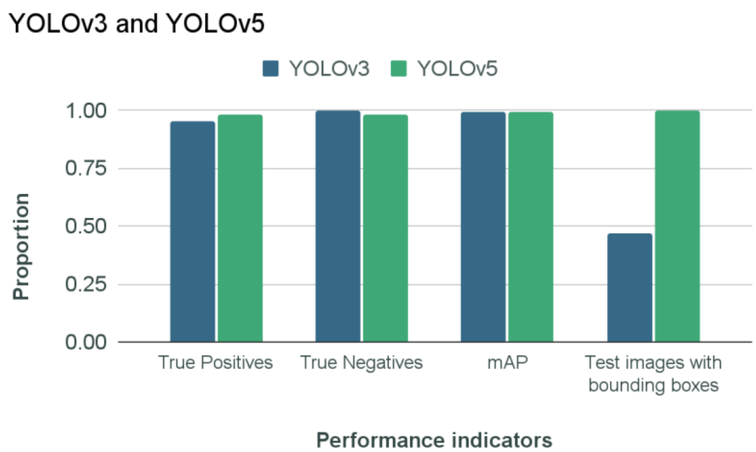


Fig. 4 Comparison of performance of YOLOv3 and YOLOv5

The results show that YOLOv5 significantly improves the performance of the ML model as the proportion of true positives, which is central to the ultimate goal to detect Aedes mosquitoes, increases when YOLOv5 is used in place of YOLOv3. Furthermore, the significant difference in proportion of test images with bboxes around them proves that YOLOv5 enhances the performance of the model as it improves the ability of the model to detect the presence of mosquitoes. Additionally, according to literature reports, YOLOv5 reduces inference speed, increases object detection accuracy, and enhances localization accuracy [22]. Therefore, it can be concluded that YOLOv5's performance surpasses YOLOv3's.

3.1.2 Dataset Size

When 200 images were used, the proportion of true positives was 1.00, while the proportion of true negatives was 0.90 and the mAP was 0.993. The proportion of test images with bboxes, meanwhile, was 100% (20 out of 20 images).

When 600 images were used, the proportion of true positives was 0.98, while the proportion of true negatives was 0.98 and the mAP was 0.994. The proportion of test images with bboxes, meanwhile, was 100% (60 out of 60 images).

When 1000 images were used, the proportion of true positives was 1.00, while the proportion of true negatives was 0.98 and the mAP was 0.994. The proportion of test images with bboxes, meanwhile, was 100% (100 out of 100 images). (See appendix for confusion matrices)

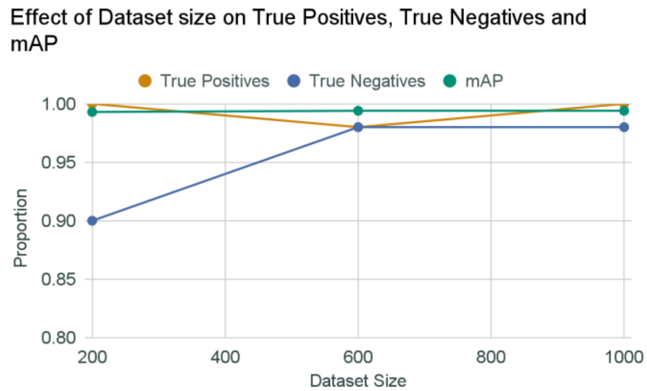


Fig. 5 Comparison of performance of models with different dataset sizes

As seen from Fig. 5, the primary difference resulting from the dataset size is the proportion of true negatives which increases with dataset size. Therefore, it can be concluded that the increase in dataset size enhances the performance of the ML model.

3.1.3 Batch Number

The number of batches tested were 8, 16, 32 and 64.

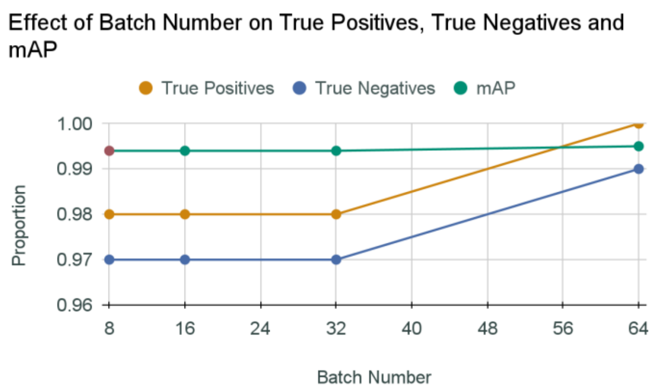


Fig. 6 Comparison of performance of model when different batch numbers are used

From Fig. 6, it can be observed that when 64 batches are used, the performance of the model is enhanced as the proportion of true positives, proportion of true negatives and mAP are increased. Nonetheless, the overall effect of batch number on performance of the model is negligible as the proportion of true positives and true negatives and the mAP remain relatively constant. (See appendix for confusion matrices)

3.1.4 Epoch Number

The effect of epoch number on performance of the model was then investigated. Specifically, the effect of decreasing the number of epochs on the performance was studied. Intuitively, increasing the number of epochs would enhance the performance of the model, hence it would be more insightful to look into the effect of decreasing the epoch number, especially since decreased epoch number reduces the training time.

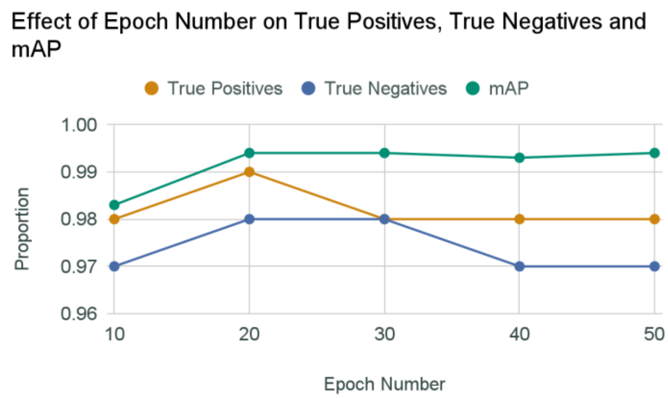


Fig. 7 Comparison of performance of model when different epoch numbers are used

From Fig. 7, it can be seen that the epoch number has no clear effect on the performance of the model, apart from the significantly lower mAP when 10 epochs are used. Therefore, if a shorter training time is the priority, epochs less than 50 but more than 10 can be used to train the model. (See appendix for confusion matrices)

3.1.5 Data Augmentation

Table 1. Results of comparison of data augmentation techniques

Augmentation technique	Number of images with no bbox	Number of inaccurate classifications	Accuracy rate
None	2 (1 unaugmented, 1 merge)	0	98.0%
Blur	6 (1 unaugmented, 3 noise, 2 merge)	1	93.0%
Noise	3 (1 unaugmented, 2 merge)	0	97.0%
Merge	2 (1 unaugmented, 1 noise)	0	98.0%
Exposure	1 (1 merge)	2	97.0%
Everything	0	0	**100.0%**

From the table, it is evident that the dataset that included all data augmentations performed the best given that all images had bboxes around them, indicating the model's ability to identify the mosquito in every image with a 100% accuracy rate. It can also be observed that models with datasets treated with only single augmentation techniques performed worse when classifying images treated with other augmentation techniques. However, these models, especially blur, performed worse than the dataset with no augmentation applied at all. A plausible explanation would be that some models with single augmentation were overfitted to images with that particular augmentation, and were thus less accurate in classifying other images. Therefore, it can be concluded that if data augmentation is to be used to improve the model, a variety of them should be used together rather than only one technique at a time.

3.2 Networking for Edge Intelligence

Since the aim of the deployment of the ML model is to track the Aedes mosquito population in real-time, it was crucial to maximise the speed of the processing of each image. To reduce the amount of time it took to make an inference on image, the YOLOv5 model was loaded before it entered a loop of sending images between the Raspberry Pi and Jetson Nano, since the model takes a substantial amount of time to load. The model used was the optimal one that was trained on a dataset that included an amalgamation of data augmentation techniques

Another consideration regarding the inference speed was where the processing of the image would be implemented. Thus, the inference speeds on the Raspberry Pi and Jetson Nano were compared by taking the mean of the inference timings of 100 images. Since the YOLOv5 inference process is split into three parts– pre-processing, inference and non-max-suppression, the overall inference timing was taken to be the sum of the time taken for these three stages.

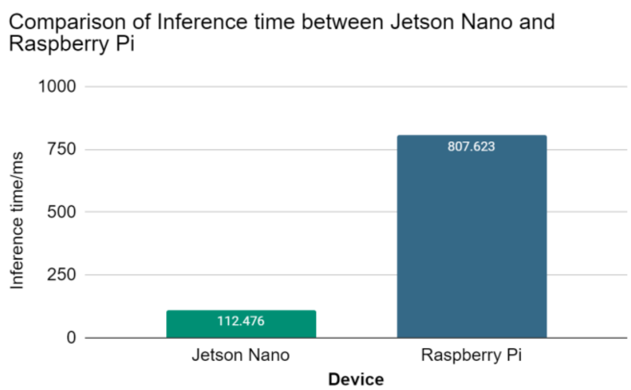


Fig. 8 Comparison of inference timing of Jetson Nano and Raspberry Pi

As evidenced by Fig. 8, the Jetson Nano takes a shorter time overall for inference. Therefore, while the images were captured on the Raspberry Pi, they were sent over a network to the Jetson Nano to be processed.

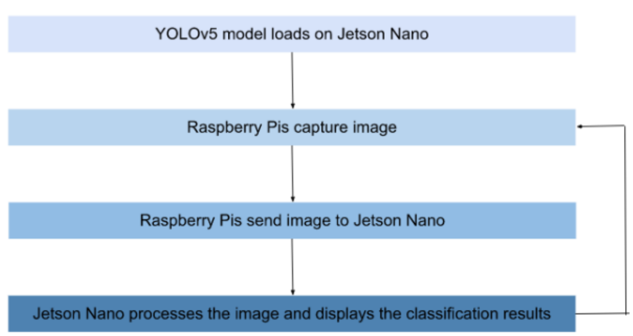


Fig. 9 Overview of Networking Pipeline

4 Conclusion

Overall, this report has investigated the effects of key data augmentation techniques, which are essential to the robust performance of the ML model in the Gravitraps operating in challenging practical conditions. This research has also found that an amalgamation of data augmentation techniques proves more effective in enhancing model performance as compared to using single augmentation techniques, and this finding has enabled the design of an optimal ML model for identification of Aedes mosquitoes.

Another key contribution of this study lies in the integration of ML with networking of the edge devices (Raspberry Pis) to the Jetson Nano, demonstrating the possibility of efficient edge deployment of ML models at Gravitraps around the country and the transfer of information back to the command centre to reduce manpower required. Overall, with the final model achieving an accuracy rate of 100% and an average 112ms inference time on the Jetson Nano, this paper presents an accurate and efficient model to track the Aedes mosquito population in Singapore.

5 Limitations and Future Work

The limitations of this research primarily concern the practical deployment of the ML model on the traps.

Future expansion of the ML model can include automating the retraining of the model with the new live images captured in deployment. By conducting online learning, the ML model can continuously upgrade itself as a sustainable model for Aedes monitoring in the long-run.

Additionally, the model currently receives images to classify. Hence, the possibility of having a 24-hour video streaming function could be investigated to constantly survey the presence of mosquitoes in the trap. Feasibility studies could be conducted regarding the power consumption of such edge devices.

Furthermore, the networking test bed makes use of a router. The Wi-Fi networking is a good proxy and a smaller testbed for this research. In practical deployment with more devices to be networked, this model can be easily transferred onto existing mobile cellular networks islandwide. The Raspberry Pis, as the edge devices, also have functionalities to enable connection to mobile networks.

Nevertheless, this research still presents pertinent contributions to the use case of the Aedes outbreak in Singapore, and it can similarly be adapted for the monitoring of other related diseases such as malaria and zika that require object detection and networking.

6 Acknowledgements

The completion of this project could not have been accomplished without the support of our mentors, Mr Teo Jenn Yue, Mr Tan Peng Loon Alex and Mr Lai Tien Leong Andrew, who have provided invaluable support and guidance throughout our research and experimental process.

7 References

[1] Ministry of Health. (2022, June 17). *Dengue.* Retrieved December 21, 2022, from https://www.moh.gov.sg/diseases-updates/dengue

[2] Centers for Disease Control and Prevention. (2021, September 20). *Dengue Symptoms and treatment.* Centers for Disease Control and Prevention. Retrieved December 21, 2022, from https://www.cdc.gov/dengue/symptoms/index.html

[3] Mohan, M. (2022, April 14). *Dengue surge fuelled by more mosquitoes, re-emergence of previously uncommon virus serotype: Experts.* Channel News Asia. Retrieved December 21, 2022, from https://www.channelnewsasia.com/singapore/dengue-surge-more-mosquitoes-uncommon-virus-serotype-2626016

[4] National Environment Agency. (2022, November 17). *Areas with higher Aedes aegypti mosquito population.* Retrieved December 21, 2022, from https://www.nea.gov.sg/dengue-zika/Aedes

[5] Government Technology Agency of Singapore. (2021, June 13). *Using data in the fight with dengue!* Youtube. Retrieved December 21, 2022, from https://youtu.be/nKbBaumLrIU

[6] Government Technology Agency of Singapore. (2021, June 13). *Using data in the fight with dengue!* Youtube. Retrieved December 21, 2022, from https://youtu.be/nKbBaumLrIU

[7] Kittichai, V., Pengsakul, T., Chumchuen, K. *et al. Deep learning approaches for challenging species and gender identification of mosquito vectors.* Sci Rep 11, 4838 (2021). Retrieved December 23, 2022 from https://doi.org/10.1038/s41598-021-84219-4

[8] Yin, M.S., Haddawy, P., Ziemer, T. *et al. A deep learning-based pipeline for mosquito detection and classification from wingbeat sounds.* Multimed Tools Appl (2022). Retrieved December 23, 2022 from https://doi.org/10.1007/s11042-022-13367-0

[9] Surya, A. *et al.* (September 16, 2022). *A Mosquito is Worth 16x16 Larvae: Evaluation of Deep Learning Architectures for Mosquito Larvae Classification.* Retrieved December 23, 2022 from https://doi.org/10.48550/arXiv.2209.07718

[10] Lo, J.-E., Kang *et al* (2021, December 28). *Data homogeneity effect in deep learning-based prediction of type 1 diabetic retinopathy.* Journal of Diabetes Research. Retrieved December 21, 2022, from https://www.hindawi.com/journals/jdr/2021/2751695/

[11] Elrahman, S. M. A., Abraham, A. (2013). *A review of class imbalance problem*. Retrieved December 21, 2022, from http://www.ias04.softcomputing.net/jnic2.pdf

[12] Redmon, J. (2016, May 9). *You Only Look Once: Unified, Real-Time Object Detection.* Retrieved December 21, 2022, from https://arxiv.org/pdf/1506.02640.pdf

[13] Jeong, H *et al.* (2018). *Image Preprocessing for Efficient Training of YOLO Deep Learning Networks.* Retrieved December 21, 2022 from 10.1109/BigComp.2018.00113.

[14] Google. (2022, December 15). *Bigquery ML Model Inference Overview.* Google. Retrieved December 21, 2022, from https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-inference-overview#:~:text=Machine%20learning%20inference%20is%20the,machine%20learning%20model%20into%20production.%E2%80%9D

[15] Subramanyam, V. S. (2021, January 28). *Non Max Suppression (NMS).* Medium. Retrieved December 21, 2022, from https://medium.com/analytics-vidhya/non-max-suppression-nms-6623e6572536

[16] Ethem, A. (2021). *Machine learning.* Amazon. Retrieved December 21, 2022, from https://docs.aws.amazon.com/machine-learning/latest/dg/improving-model-accuracy.html

[17] *What does "background" Fn and fp in the confusion matrix of Yolov7 stand for*. Stack Overflow. Retrieved December 21, 2022, from https://stackoverflow.com/questions/74016101/what-does-background-fn-and-fp-in-the-confusion-matrix-of-yolov7-stand-for
[18] Brownlee, J. (2020, September 18). *Hyperparameter optimization with random search and grid search*. MachineLearningMastery.com. Retrieved December 21, 2022, from https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/.
[19] *Are heat waves destroying your image quality?* The Digital Picture. Retrieved December 21, 2022, from https://www.the-digital-picture.com/Photography-Tips/Heat-Waves-and-Photography.aspx
[20] *Socket programming*. IBM. (2021, August 31). Retrieved December 22, 2022, from https://www.ibm.com/docs/en/i/7.1?topic=communications-socket-programming
[21] Kalita, L. (2014). *Socket Programming.* International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 4802-4807. Retrieved December 22, 2022, from https://doc.presentica.com/11477009/5ebad98adffc8.pdf
[22] Nepal, Upesh & Eslamiat, Hossein. (2022). *Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs.* Sensors. 22. 10.3390/s22020464.

Appendix

Examples of Dataset Images

	
Example of image from Kaggle dataset	Example of image from the Internet
	
Example of image from Dryad dataset	Example of image from the Internet

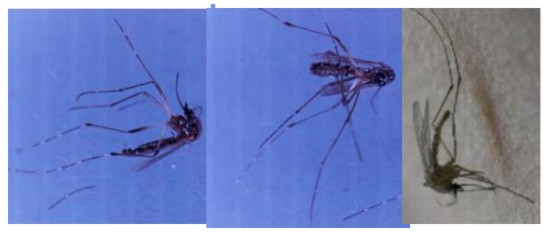
Examples of Blurred Images

	
Example of blurred image of Aedes mosquito	Example of non-blurred image of Aedes mosquito

Examples of Images with Noise Added

	
Example of image of Aedes mosquito with noise added	Example of image of Aedes mosquito with no noise added

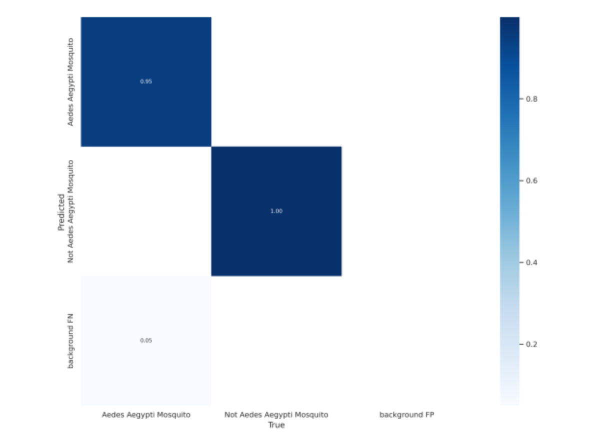
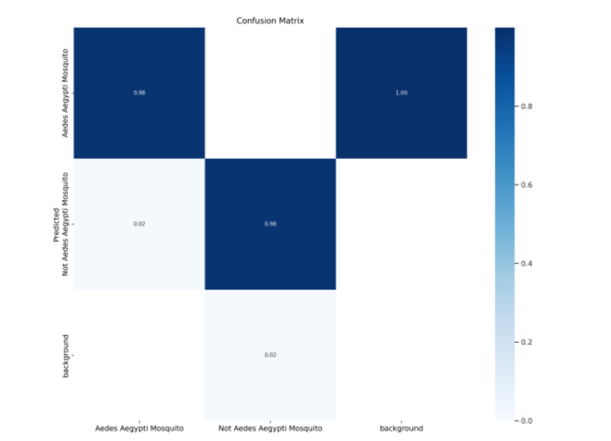
Examples of Merged Images

	
Example of merged image of 1 Aedes mosquito and 1 non-Aedes mosquito	Example of merged image of 2 Aedes mosquitoes and 1 non-Aedes mosquito

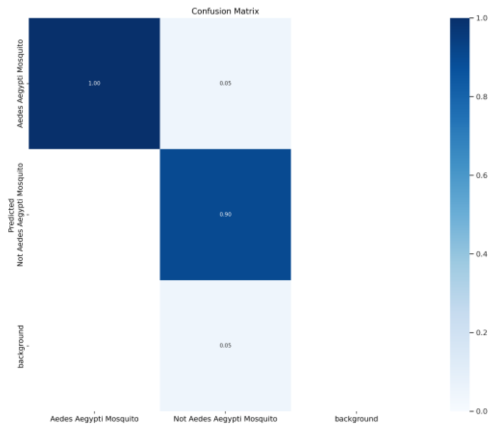
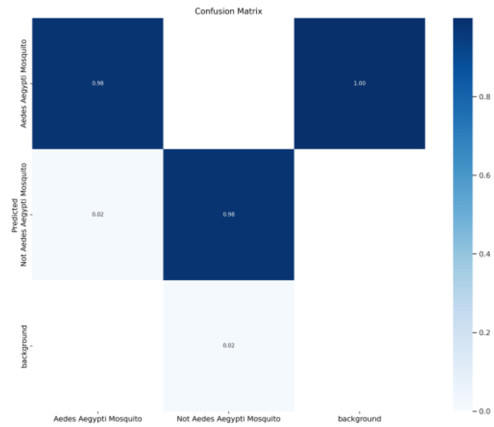
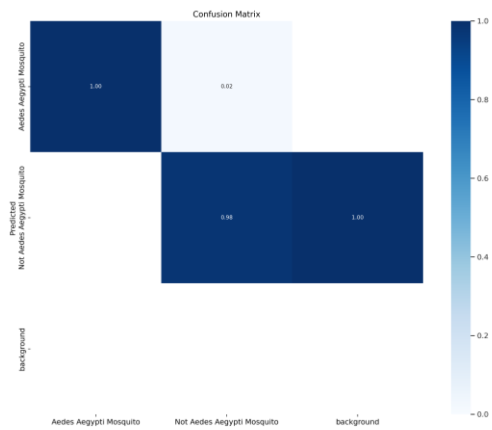
Examples of Images with Altered Exposure

	
Example of image of Aedes mosquito with increased exposure	Example of image of Aedes mosquito with non-altered exposure
	
Example of image of Aedes mosquito with decreased exposure	Example of image of Aedes mosquito with non-altered exposure

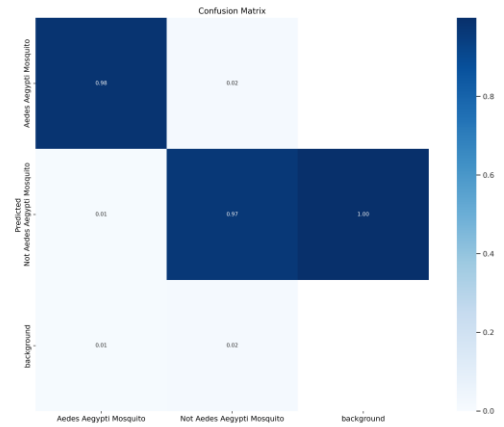
Confusion Matrices Obtained when Comparing YOLO Version

	
Confusion matrix when YOLOv3 was used (600 image dataset)	Confusion matrix when YOLOv5 was used (600 image dataset)

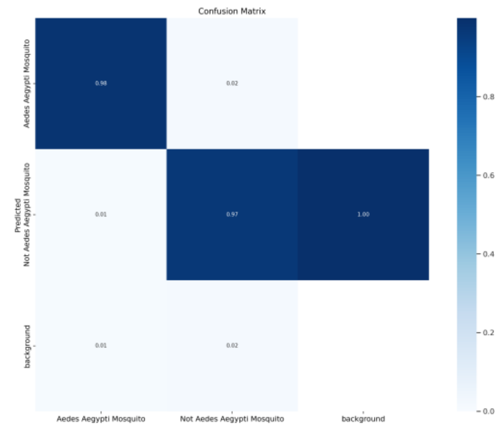
Confusion Matrices Obtained when Comparing Dataset Size

<table>
<tr><td>
</td><td>
</td></tr>
<tr><td>Confusion matrix when 200 images were used</td><td>Confusion matrix when 600 images were used</td></tr>
<tr><td colspan="2">
</td></tr>
<tr><td colspan="2">Confusion matrix when 1000 images were used</td></tr>
</table>

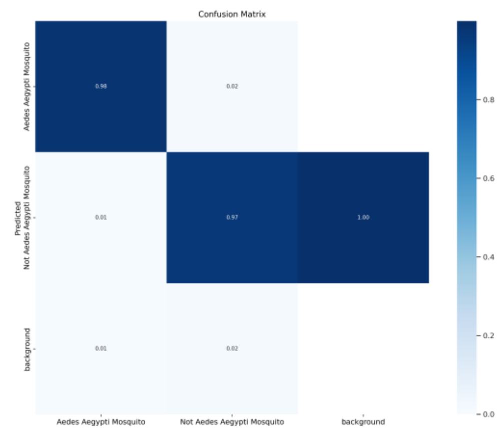
Confusion Matrices Obtained when Batch Numbers are Varied



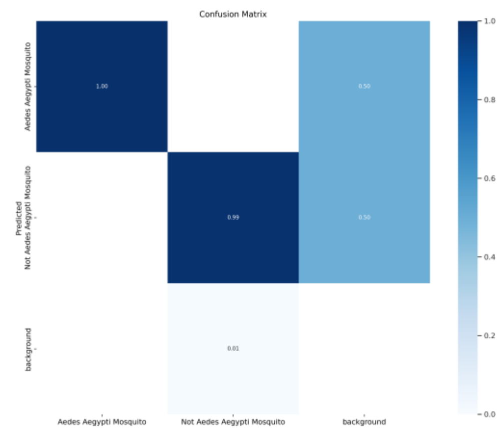
Confusion matrix when 8 batches are used



Confusion matrix when 16 batches are used

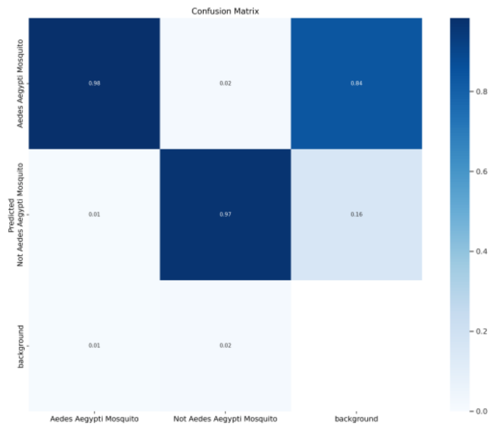


Confusion matrix when 32 batches are used

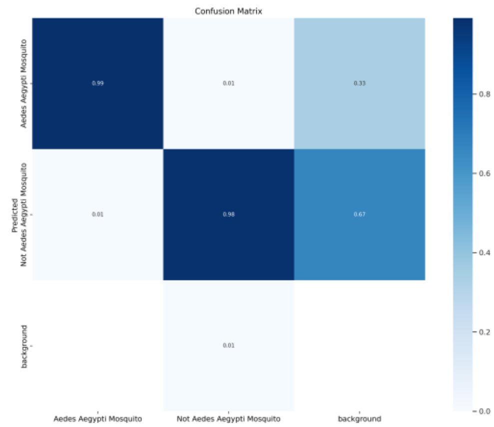


Confusion matrix when 64 batches are used

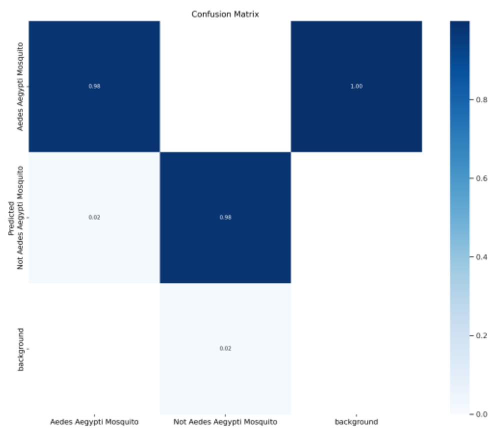
Confusion Matrices Obtained when Epoch Number is Varied



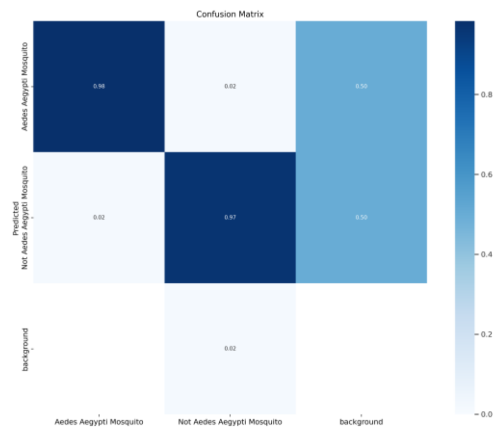
Confusion matrix when 10 epochs are used



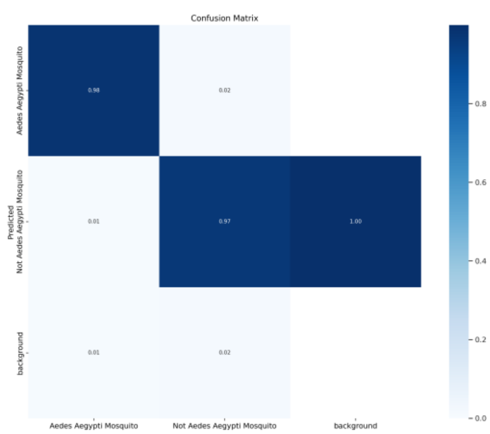
Confusion matrix when 20 epochs are used



Confusion matrix when 30 epochs are used



Confusion matrix when 40 epochs are used



Confusion matrix when 50 epochs are used

Data

Mosquito Classification	Data Source
Aedes	Kaggle: https://www.kaggle.com/datasets/pradeepisawasan/Aedes-mosquitos
Non-Aedes	Dryad: https://doi.org/10.5061/dryad.z08kprr92