A YOLO-Based Approach for Aedes Aegypti Larvae Classification and Detection

Angelo M. Hubalde
School of Electrical, Electronics, and Computer Engineering
Mapúa University
Manila, Philippines
amhubalde@mymail.mapua.edu.ph

Dionis A. Padilla
School of Electrical, Electronics, and Computer Engineering
Mapúa University
Manila, Philippines
dapadilla@mapua.edu.ph

Denzel Antoinette C. Santos
School of Electrical, Electronics, and Computer Engineering
Mapúa University
Manila, Philippines
dacsantos@mymail.mapua.edu.ph

2021 6th International Conference on Image, Vision and Computing (ICIVC) | 978-1-6654-4368-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICIVC52351.2021.9527017

***Abstract*—Mosquitoes are small, flying insects that can be found almost anywhere in the world. Although most of these creatures are just considered a nuisance, a few of its species can be vectors of some of the deadliest viruses and diseases that can greatly affect living organisms—animals and people alike. Aedes aegypti species, for instance, are deemed as the primary vector of the infamous dengue virus that has been around for a long time. With this, there have been many studies, mostly only with limited success, in different fields of learning on how to control and suppress both the mosquitoes and the diseases they carry. While some of these studies and research experiments are geared towards the use of modern technology such as machine learning, deep learning, and image processing, there are still a lot of strategies and methods that are yet to be discovered to improve present-day approaches and attain better results that are more sustainable and efficient. And so, the main goal of this study is to aid in the early suppression and prevention of the primary vector of dengue by creating a system implemented with the You Only Look Once (YOLO) algorithm that can detect and classify whether a mosquito larva is an Aedes aegypti species or not. Results showed that the system was able to correctly recognize most of the larval samples subjected to testing, with a root mean square error of 0.45 and 0.77 for Aedes and Non-Aedes larvae respectively.**

Keywords—aedes aegypti, vectors, dengue, deep learning, YOLO algorithm

I. Introduction

Mosquitoes are without a doubt considered the world's most dangerous creatures. It might seem impossible that something so small could be deadlier than most animals, but these insects carry and spread diseases capable of taking the most lives, killing more people in a day than larger organisms do in a period of a hundred years. According to World Health Organization, mosquito-borne diseases cause the death of about a million people every year [1]. Malaria— one of the well-known serious diseases spread by the bite of an infected mosquito— caused most of these deaths and leaves people at risk with the growing number of cases every year. In the Philippines alone, this specific disease puts approximately 14 million people at risk especially those living in remote areas where there is little to no access at all to healthcare services. Aside from malaria, some other known mosquito-borne diseases making headlines today and continuously putting public health at risk are the: Zika virus, chikungunya virus, Japanese encephalitis, and dengue— the world's fast-emerging virus— that has greatly affected over a hundred thousand Filipino people in 2017 [2] and put the country under the "national dengue alert" declaration in 2019 because of the surge in the number of cases reported in just six months [3]. The known primary vectors of dengue and other mosquito-borne diseases mentioned above are the Aedes aegypti mosquitoes— commonly named as yellow-fever mosquitoes— present in almost every urban area there is [4]. With this, health authorities in the Philippines persistently take part in devising ways and strategic initiatives to eradicate the risk of the aforementioned viral infections.

Some of the conventional ways of preventing the spread of these viruses, especially dengue, include manual destruction of breeding places, use of repellents, and immediate consultation from health specialists when the first signs and symptoms of the infection strike [5]. Generally, these flying insects are minuscule and often hide in areas that are inhabited, not well-maintained making it difficult to situate traps and for fogs and pesticides to permeate. Although a preventive measure is represented in these methods, better and more advanced ways are now being introduced to do early suppression, targeting only the ones that bear the most responsibility for the transmission of the infections. In this way, affected species are minimized and efficiency is maximized. Compared to the imago stage of mosquitoes, its larvae tend to dwell, gather, and grow in pools of water such as puddles, which can be found almost everywhere especially in most tropical countries. Ergo, controlling mosquito larvae is easier and much more efficient than when they are at the imago stage [6] especially now that are new and innovative ways to address the issues on virus-carrying mosquitoes. Many have attempted and are still making an effort to establish projects and studies to help prevent, if not completely stop, the outbreak of mosquito-borne diseases. Correspondingly, there have been numerous research experiments, although mostly with limited success, on strategies to counteract and control the population of mosquitoes and the spread of the viral infections they carry from different points of view. Currently, in the field of computing, techniques using modern technology such as image processing are being widely used to detect and control adult mosquitoes. A study on a Raspberry Pi-based medical system to pre-diagnose mosquito-borne diseases was proposed by [7]. In another work, the researchers developed an algorithm and a device that can detect blood parasites from blood smears [5]. The utilization and incorporation of machine learning with image processing is now also being practiced for most of the results show efficiency and good performance. For instance in 2016, [8] proposed a way to detect whether an adult mosquito is an Aedes aegypti or not using the Support Vector Machine (SVM) algorithm. Another group of researchers used a deep-learning-based method— by means of Convolutional Neural Network (CNN)— of classifying mosquito larvae to know which ones are to be fumigated afterward to discontinue the growth of the virus-carrying mosquitoes. In this way, the process of identifying the infected larvae is made faster and more efficient than the conventional methods, thus decreasing the rate of infection [9]. CNNs are of broad and current interest because of its significant contribution in improving various computer vision methodologies [10]–[12]. With its capabilities, it has already been used in a variety of practical applications, such as the localization of pests in agricultural fields [13] and insect image recognition [14] among others. From this, many other techniques utilizing CNNs have emerged and the You Only Look Once (YOLO) algorithm is one. From its name, using a single CNN, it looks at an image once and then predicts what objects are existent and where each of these objects are [15].

With the world continuously becoming more technologically advanced, the traditional ways, requiring more time and effort, to prevent the spread of diseases carried out by mosquitoes such as dengue are now far behind the fast-paced environment, making it less effective and tedious to implement. The previously mentioned classic approaches generally only focus on suppressing and controlling adult mosquitoes. In spite of the many studies being established these days, there are still a lot of techniques and approaches that are yet to be discovered to improve the schemes of existing research experiments and achieve far greater outcomes using mosquito larvae as the key subject in the early suppression of mosquito-borne diseases especially dengue.

The main goal of this study is to classify whether a mosquito larva is an Aedes aegypti species or not, to aid in the early suppression and prevention of the primary vector of dengue. To achieve this, the following specific objectives must first be taken into consideration: Implement the YOLO algorithm for mosquito larvae classification and detection; Develop a prototype device that can capture and classify Aedes aegypti mosquito larvae from other mosquito larvae using NVIDIA Jetson Nano and analyze the system's accuracy using the Root Mean Square Error.

The proposed study would make classification and detection of virus-carrying mosquitoes faster and much more efficient than conventional methods thus, will greatly help in preventing and suppressing the spread of mosquito-borne diseases at an earlier phase. Additionally, the system will be made portable so it's easier to incorporate other systems. This will further aid researchers and organizations in amplifying their control strategies. In terms of technology utilization, this study will showcase how well the YOLO algorithm performs especially in small-scale fields and objects so, this could be a good avenue to assess its capabilities and what areas need improvement.

II. Methodology

Dengue is one of the well-known mosquito-borne viruses that pose a serious threat to the general public's health, especially to those residing in countries with tropical climates [16]. Vector control is one of the major focuses of some research experiments for this strategy is known to be much more efficient than other methods because there have been no vaccines or specific treatments being developed yet that are effective enough to treat most mosquito-borne diseases [17]. And as modern technology continues to revolutionize the world, there are now many more innovative ways to control these vectors and suppress the diseases they transmit, such as the utilization of deep learning techniques [18]. In particular, CNN and YOLO recently gained attention on the topic of object recognition [19]–[23]. Since YOLO's introduction, newer versions and updates of it have emerged and have been used in a variety of domains, among which were in establishment and road surveillance technologies [24]–[26], medical research [27], and robotics [28].

This paper will focus on the detection and classification of Aedes aegypti larvae using the tiny model of YOLOv3, implemented on an NVIDIA Jetson Nano equipped with a high-resolution camera. While YOLOv3 is already a powerful algorithm with multi-scale predictions and a performance better than other schemes [29], it is still a significantly large model especially for a resource-constrained setup such as this study. Since the proposed system demands real-time application as well by considering both image and video stream inputs, the researchers opted to use the YOLOv3-Tiny model. Although accuracy is slightly cut down in this version of YOLO, its processing speed is relatively faster [30].

The conceptual framework of the system is as shown in Fig. 1. It will start by acquiring images and video streams taken using a camera or from an online source. These inputs will then undergo processing for detection and classification using the YOLOv3-Tiny algorithm. The system's evaluation will only be whether the images are to be labeled as "Aedes" for Aedes aegypti or "Non-Aedes" for others. The species of mosquito larvae that will be subjected to training and testing will only be limited to what is available. Moreover, the gender of the Aedes aegypti larvae will not be identified.

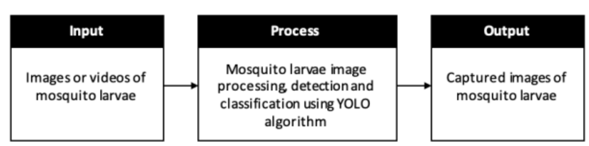


Fig. 1. Conceptual Framework.

A. Hardware Design

Fig. 2 shows the basic hardware block diagram of the system. Initially, a camera will feed images or video sequences of mosquito larvae to NVIDIA Jetson Nano which in turn, will process the acquired data for object detection and classification using the YOLO algorithm. Once completed, the results will then be displayed on the LCD screen. For image inputs, the system count of both Aedes and Non-Aedes will also be displayed.

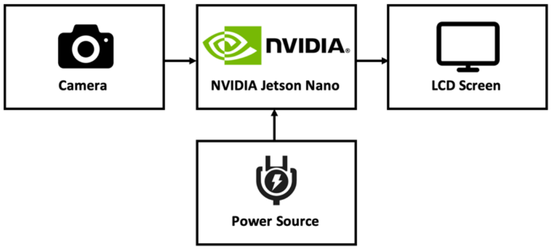


Fig. 2. Hardware Block Diagram.

B. Data Gathering

Actual mosquito larvae used in this study were collected from one of mosquitoes' usual breeding grounds— an old container left with stagnant water— and were sent to the Research Institute for Tropical Medicine (RITM) for identification. Images and videos were then captured using either a smartphone camera at 4x zoom or a custom-built acquisition system comprised of a digital microscope with up to 1000x magnification and a tray to hold the larvae as shown in Fig. 3 below. Aside from actual mosquito larvae, the dataset used in this study also includes images and videos taken from various web sources.

(a) Actual setup (b) Actual image captured

Fig. 3. Image Acquisition System.

C. System Development

A custom dataset comprised of two classes, Aedes and Non-Aedes, was used in this study. Images for each class underwent augmentation— flipping and cropping among others— to enlarge the dataset and were then labeled accordingly using the annotation tool LabelImg. After the initial setup of preparing the dataset and necessary configuration files, the system was then trained and tested using the tiny model of the YOLOv3 algorithm— developed specifically for constrained environments— with Darknet as the framework. YOLOv3-Tiny, as its name suggests, is a scaled-down, simplified version of the YOLOv3 model that has lesser layers and parameters, but still uses the same batch normalization method and leaky ReLU function [31]. While there is a trade-off between inference speed and accuracy [30], this tiny model nevertheless delivers good performance compared to other traditional computer vision techniques [32], ideal for embedded devices [33] and systems with real-time requirements and few resources [34].

The main program flowchart of the Aedes aegypti larvae classification and detection system is as shown in Fig. 4. It starts by getting larval images or video sequences and then pre-process it to fit the required image size, in this case, 416 x 416, that will be sent for processing by the YOLO algorithm. Subsequently, obtained data are labeled in accordance with the system requirements. Other processes such as feature extraction, object detection, and classification will be done using the tiny model of the YOLOv3 algorithm. To remove the unnecessary bounding boxes around the targeted objects, thresholding and non-maximum suppression will also be implemented before displaying the final detected and classified larval images to the monitor.

Non-maximum suppression starts by thresholding as shown in Fig. 4. It first determines whether the predicted bounding boxes have an objectness score greater than the threshold. Those with scores greater than the ceiling will be ignored otherwise retained. Once done, it will then determine if there are any overlapping bounding boxes left. If there are, the one with the highest objectness score must be selected while those with scores greater than or equal to the threshold must be removed until overlapping bounding boxes are not evident anymore.

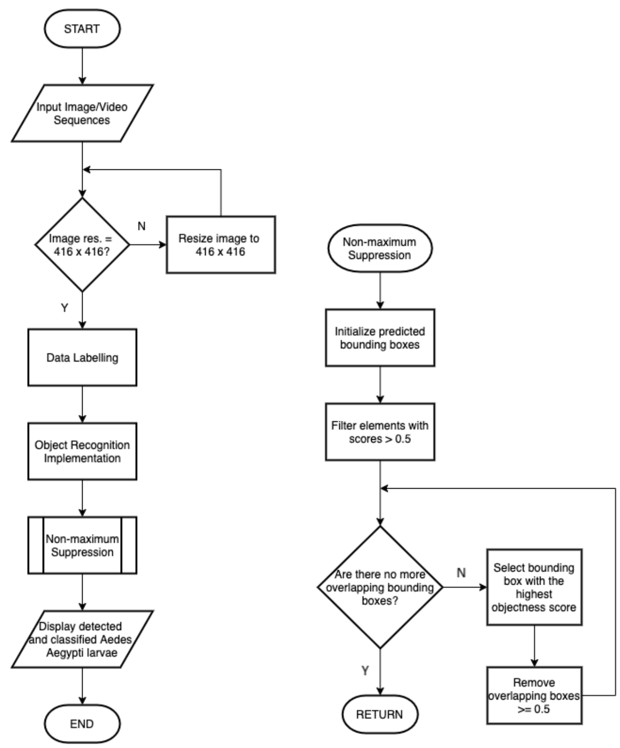


Fig. 4. Main Program and Non-maximum Suppression Flowcharts.

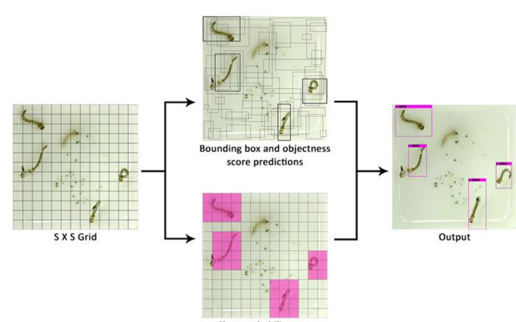


Fig. 5. YOLOv3-Tiny's process implemented on the system.

Unlike YOLOv3-Tiny's predecessor that uses Darknet53 for its architecture, this simplified version uses the Darknet19 structure which is relatively streamlined for studies with limited hardware resources but has real-time requirements. Shown in Fig. 5 is the graphical representation of YOLOv3-Tiny's process. Here, input images are divided into S x S grids— in this case at two feature map scales, 13 x 13 and 26 x 26— that predict the bounding boxes, confidence scores, and class probabilities until a final result is generated.

The training went on for about an hour or two with 4000 iterations and stopped with an average loss of 0.8094, which is already a fair figure for the custom and relatively small dataset of this study. All this was deployed on an NVIDIA Jetson Nano since the system required real-time application and fast processing speed. This development board was chosen to run the system because of its much better processing capability to carry out modern AI and machine learning workloads, especially with its 128-core GPU that is more than enough to manage real-time applications [35]. Moreover, the system will accept inputs from the embedded camera for real-time processing where recognized Aedes aegypti larvae will be bounded by pink boxes, otherwise green.

D. *System Testing*

A total of 173 mosquito larvae and 4 pupae distributed into 25 groups were used to test the system. Due to resource constraints, only 35 of the larval samples were of different species belonging to the genus Culex and Anopheles while the rest of the samples were already Aedes aegypti. Furthermore, combining Aedes aegypti with the other species in one container and video was not possible at the time of the testing, so a total of 25 trials were done— the first 12 trials with 4-15 mosquito larvae each were for Aedes aegypti and the other 13 trials with 1-10 larvae each were for the other species. In each trial, samples were filmed for 10-20 seconds to make sure all were recognized and seen in the frames.

III. Results and Discussion

Table I shows both the system and actual counts of the mosquito larvae from video inputs. Out of the 138 actual Aedes aegypti and 35 Non-Aedes larval samples, the system correctly identified all larvae but one as Aedes in the first 12 trials. Since video inputs were used in this testing portion of the system, the larvae count for each frame varied and went on until the duration of the video ended. While larval samples were correctly classified in most of the videos, there were some inaccuracies such as the ones presented in video numbers 6, 11, and 12, identifying Aedes aegypti pupae as Aedes. This happens if the pupa has just recently emerged from its 4th instar larval stage and still slightly resembles some of the physical attributes of a larva, especially when it stretches from its usual curled up position.

To further analyze, frames from one of the testing video outputs were extracted. Shown in Fig. 6 are Video #11's frames 126 and 179 respectively at 10 fps. The pupa in this video, encircled in red, has just on the spot developed from its 4th instar stage with its old larval skin, encircled in blue, still evidently seen in the frame. In this context, its movement and some of its physical characteristics still vaguely bear a resemblance to that of a larva in the 4th instar. In this specific testing video, out of its 200 video frames, the pupa was only detected and classified as Aedes in the 7 frames it was wriggling. Unlike the larvae, the system did not incessantly recognize the pupa for the duration of the video. Moreover, there were several times in some video frames where the system collectively recognized the larvae. This happens when there is crowding of samples in one area especially in trials with comparatively more larvae such as those with 15 larval samples.

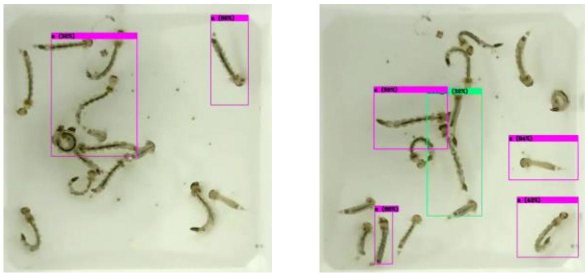
TABLE I. Testing Results for Aedes and Non-Aedes Larval Samples

Video #	Aedes		Non-Aedes	
	System Count	*Actual Count*	*System Count*	*Actual Count*
1	15	15	1	0
2 (with 1 pupa)	15	15	0	0
3	15	15	0	0
4	15	15	0	0
5	15	15	0	0
6 (with 1 pupa)	16	15	0	0
7	10	10	0	0
8	10	10	0	0
9	10	10	0	0
10	9	9	0	0
11 (with 1 pupa)	5	4	0	0
12 (with 1 pupa)	6	5	0	0
13	0	0	1	1
14	1	0	0	1
15	0	0	1	2
16	0	0	2	2
17	0	0	0	1
18	0	0	1	1
19	0	0	0	1
20	0	0	10	10
21	1	0	0	2
22	0	0	0	2
23	0	0	4	5
24	0	0	5	6
25	0	0	1	1

(a) Frame 126 (b) Frame 179

Fig. 6. Video frames from testing video#11.

Fig. 7 below shows two of the video frames from trial number 1. In Frame 192, on the left side, there is an apparent crowding of larvae that the system classified as a group. Unlike the individually recognized larva on the top right corner with an 89% confidence score, the clustered larvae were predicted with a relatively lower score.



(a) Frame 192 (b) Frame 77

Fig. 7. ideo frames from testing video #1.

From the same previous video but a different frame, Frame 77, the system misclassified one of the Aedes aegypti larvae in Frame 77 of Video #1 as Non-Aedes, bounded by the green box and presented in Table I. This was only evident in one out of the 200 video frames of trial number 1 at 10fps where there are noticeable larvae huddling in the middle.

IV. Statistical Analysis

To assess how well the system performed, the Root Mean Square Error (RMSE) values of each class were evaluated to determine how close or far apart the predicted values are from the observed values. Using (1), the computed RMSE for the class Aedes was 0.45 and 0.77 for the class Non-Aedes.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} (P_i - O_i)^2}{N}} \qquad (1)$$

Breaking down (1), RMSE is taking the square root of the summation of the squared residual values over the total number of observations (N), where the residuals are the differences between the predicted (P_i) and observed values (O_i). While there is really no fixed threshold for RMSE since it will depend on the objective of the system being analyzed, a figure that has a comparatively low value is much preferred for it suggests a better fit, having a lower measure of the residuals between the predicted and observed values. From the obtained results, it shows that the recognition system is generally a good fit in recognizing the larval samples. Taking into account the individually measured RMSE values of each class, the system predicts the Aedes class better than the Non-Aedes with a much smaller relative error in between predictions, as demonstrated in its relatively lower RMSE of 0.45.

To visually analyze the results, Fig. 8 shows the graphical representation of the obtained RMSE value of 0.45 for the Aedes class. As shown in the scatter plot, the predicted values are concentrated around the line of best fit— obtained using the slope and intercept from the predicted and actual responses. To further support this, its R-squared or coefficient of determination was also measured using the RSQ function in Microsoft Excel to determine its goodness of fit on the regression line, where a 0 R-squared value would mean that the system is not a good fit and a value of 1 would mean it is a perfect fit. With this, the computed R-squared value for the Aedes class was 0.996 indicating that while it is not perfect, it is a value close to 1, therefore is a good fit and it can nevertheless accurately predict Aedes larval samples.

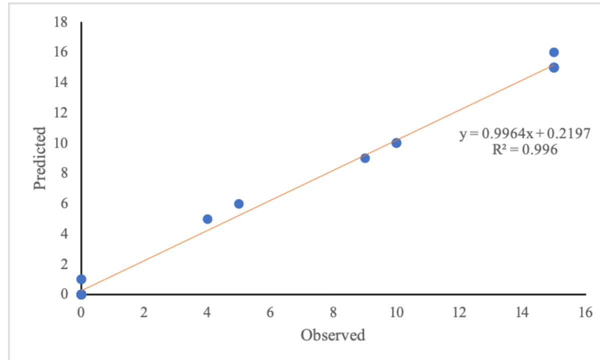


Fig. 8. Predicted vs. Observed Values of Aedes Larval Samples.

In a similar manner, the graphical representation of Non-Aedes' RMSE value of 0.77 is shown in Fig. 9. Here, while more variance is noticeable, predicted values are all the same close to the actual values. And with its R-squared value of 0.9122, it also suggests that predictions are accurate and it is a good fit as well.

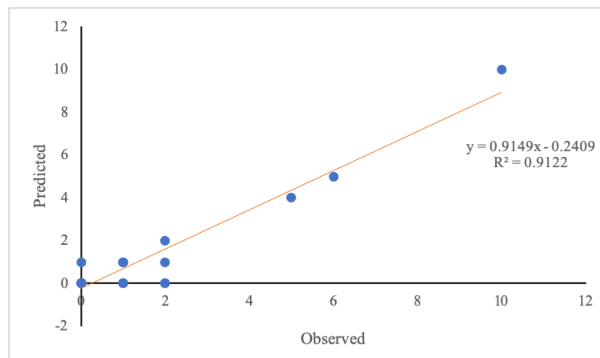


Fig. 9. Predicted vs. Observed Values of Non-Aedes Larval Samples.

Furthermore, the Normalized Root Mean Square Error (NRMSE) values of each class were also computed to measure the overall deviations between the predicted and actual values. Using (2), the computed NRMSE for Aedes and Non-Aedes were 0.08 and 0.55 respectively.

$$\mathrm{NRMSE} = \frac{\mathrm{RMSE}}{\overline{O_i}} \qquad (2)$$

The computed NRMSE backs the obtained RMSE and R-squared values by showing that the system is a good fit in identifying mosquito larval samples and that between the two classes, there is relatively less variance in the Aedes class than in the Non-Aedes class.

V. Conclusion

From the assessment presented above, it can be established that both the general and specific objectives were met. A mosquito larvae recognition system using the YOLOv3 algorithm running on an NVIDIA Jetson Nano was developed. Since one of the system's functionalities is to capture real-time videos of actual larval samples without heavily affecting inference speed and accuracy, the tiny model of the YOLOv3 algorithm was utilized. From this, an average of 6.8 frames per second (FPS) was obtained when processing video streams— a relatively good figure, especially for a resource-constrained setup. The system was able to successfully detect and distinguish Aedes aegypti larvae apart from non-Aedes species as exhibited in its RMSE analysis of 0.45 and 0.77 respectively indicating its goodness of fit.

Despite the fact that all objectives were met, the system still had crucial points such as in trials wherein it inaccurately classified pupal samples as larvae, recognized larval samples en masse, flickering predictions, and misclassified some of the non-Aedes species as Aedes. In most cases, these inaccuracies were unavoidable because the samples are fast movers, also deemed as wrigglers, their position and form changes in every video frame. Another contributing factor as to why these instances transpired was the dynamics of the camera such as its light levels and saturation of the frames captured. Moreover, the disparity between the RMSE values, although indicating a good fit, was also driven by the fact that there is a noticeable imbalance between the number of Aedes and Non-Aedes larval samples used to test the system.

VI. Recommendations

While the system was able to meet all of its intentions, general and specific, there are still lots of room for improvement especially if it will be incorporated with other, diverse methods and studies. For future work, the system may be significantly furthered with more datasets and iterations in training to yield better and more accurate outcomes. A balanced testing data as well is highly recommended to generate fairer analysis. In terms of the larval samples, it will be a considerable addition to the system's functionalities if it can classify the gender of the larvae as well, to set the female— capable of feeding on human blood to produce eggs, therefore considered the primary vectors [36] — apart from the male larvae and display the count of the recognized ones even for video inputs. Lastly, since YOLO's introduction, it has already emerged so far with much better and capable versions that can be utilized in a broad array of AI and machine learning applications. With this, newer models can be employed and analyzed in comparison to other approaches.

References

[1] World Health Organization, "Global Vector Control Response 2017-2030," Geneva, Switzerland, 2017.

[2] F. V. Salazar and K. A. Gimutao, "The Evolution of Entomological Research with Focus on Emerging and Re-emerging Mosquito-Borne Infections in the Philippines," *Public Heal. - Emerg. Re-emerging Issues*, no. October, 2018.

[3] "DOH DECLARES NATIONAL DENGUE EPIDEMIC," 2019. [Online]. Available: https://www.doh.gov.ph/press-release/DOH-DECLARES-NATIONAL-DENGUE-EPIDEMIC.

[4] D. Motta *et al.*, "Application of convolutional neural networks for classification of adult mosquitoes in the field," *PLoS One*, vol. 14, no. 1, pp. 1–18, 2019.

[5] R. G. Garcia *et al.*, "Modification of Optical Compound Microscope to Detect Malaria Blood Parasites Using Wright-Giemsa Staining Process with SURF Based Stitching," in *Proceedings of the 2019 9th International Conference on Biomedical Engineering and Technology - ICBET 19*, 2019, pp. 125–129.

[6] K. Kim, J. Hyun, H. Kim, H. Lim, and H. Myung, "A Deep Learning-Based Automatic Mosquito Sensing and Control System for Urban Mosquito Habitats," *Sensors (Basel).*, vol. 19, no. 12, 2019.

[7] G. V. Magwili, M. A. E. Latina, F. I. C. Miguel, J. K. P. Ortega, T. K. L. Pastoril, and E. J. D. Tanglao, "Raspberry pi-based medical expert system for pre-diagnosis of mosquito-borne diseases," *2018 IEEE 10th Int. Conf. Humanoid, Nanotechnology, Inf. Technol. Commun. Control. Environ. Manag. HNICEM 2018*, pp. 1–6, 2019.

[8] A. M. M. De Los Reyes, A. C. A. Reyes, J. L. Torres, D. A. Padilla, and J. Villaverde, "Detection of Aedes Aegypti mosquito by digital image processing techniques and support vector machine," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, pp. 2342–2345, 2017.

[9] A. Sanchez-Ortiz *et al.*, "Mosquito larva classification method based on convolutional neural networks," *2017 Int. Conf. Electron. Commun. Comput. CONIELECOMP*, pp. 4–9, 2017.

[10] T. Guo, J. Dong, H. Li, and Y. Gao, "Simple Convolutional Neural Network on Image Classification," in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, 2017, pp. 721–724.

[11] F. R. G. Cruz, C. J. R. Santos, and L. A. Vea, "Classified Counting and Tracking of Local Vehicles in Manila Using Computer Vision," in *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management, HNICEM 2019*, 2019.

[12] M. J. Macawile, V. V. Quiñones, A. Ballado, J. Dela Cruz, and M. V. Caya, "White blood cell classification and counting using convolutional neural network," *2018 3rd Int. Conf. Control Robot. Eng. ICCRE 2018*, pp. 259–263, 2018.

[13] R. Li *et al.*, "An Effective Data Augmentation Strategy for CNN-Based Pest Localization and Recognition in the Field," *IEEE Access*, vol. 7, pp. 160274–160283, 2019.

[14] M. Martineau *et al.*, "Effective Training of Convolutional Neural Networks for Insect Image Recognition," in *19th International Conference, ACIVS*, 2018.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[16] K. A. Agrupis, M. Ylade, J. Aldaba, A. L. Lopez, and J. Deen, "Trends in dengue research in the Philippines: A systematic review," *PLoS Negl. Trop. Dis.*, vol. 13, no. 4, pp. 1–18, 2019.

[17] A. A. de los Reyes and J. M. L. Escaner, "Dengue in the Philippines: Model and analysis of parameters affecting transmission," *J. Biol. Dyn.*, vol. 12, no. 1, pp. 894–912, 2018.

[18] P. Zhou, J. Liu, X. Liu, Z. Yang, and J. Grundy, "Is deep learning better than traditional approaches in tag recommendation for software information sites ?," *Inf. Softw. Technol.*, vol. 109, no. January 2018, pp. 1–13, 2019.

[19] M. V. C. Caya, G. V. Magwili, D. L. Agulto, R. J. Laranang, and L. K. G. Palomo, "Supervised machine learning-based fall detection," in *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2018*, 2018.

[20] M. V. C. Caya, A. N. Yumang, J. V. Arai, J. D. A. Ninofranco, and K. A. S. Yap, "Human Activity Recognition Based on Accelerometer Vibrations Using Artificial Neural Network," in *2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management, HNICEM 2019*, 2019.

[21] P. Sismananda, M. Abdurohman, and A. G. Putrada, "Performance Comparison of Yolo-Lite and YoloV3 Using Raspberry Pi and MotionEyeOS," in *2020 8th International Conference on Information and Communication Technology (ICoICT)*, 2020, pp. 1–7.

[22] E. L. Villanueva and N. B. Linsangan, "Determination of pupillary distance using YOLO algorithm," in *12th International Conference on Digital Image Processing (ICDIP)*, 2020.

[23] C. O. Tayo, N. B. Linsangan, and R. V. Pellegrino, "Portable Crack Width Calculation of Concrete Road Pavement Using Machine Vision," in *11th IEEE International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 2019.

[24] P. Ren, L. Wang, W. Fang, S. Song, and S. Djahel, "A Novel YOLO-based Real-time People Counting Approach," in *2017 International Smart Cities Conference (ISC2)*, 2017, pp. 1–2.

[25] W. Lan, J. Dang, Y. Wang, and S. Wang, "Pedestrian Detection Based on YOLO Network Model," in *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2018, pp. 1547–1551.

[26] Z. Rahman, A. M. Ami, and M. A. Ullah, "A Real-Time Wrong-Way Vehicle Detection Based on YOLO and Centroid Tracking," in *2020 IEEE Region 10 Symposium (TENSYMP)*, 2020, pp. 916–920.

[27] Y. Nie, P. Sommella, M. O'Nils, C. Liguori, and J. Lundgren, "Automatic Detection of Melanoma with Yolo Deep Convolutional Neural Networks," in *2019 E-Health and Bioengineering Conference (EHB)*, 2019, pp. 1–4.

[28] J. Y. Yang, U. K. Chen, K. C. Chang, and Y. J. Chen, "A Novel Robotic Grasp Detection Technique by Integrating YOLO and Grasp Detection Deep Neural Networks," in *2020 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, 2020, pp. 1–4.

[29] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018.

[30] P. Adarsh, P. Rathi, and M. Kumar, "YOLO v3-Tiny: Object Detection and Recognition using one stage improved model," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020, pp. 687–694.

[31] R. Gopal, S. Kuinthodu, M. Balamurugan, and M. Atique, "Tiny Object Detection: Comparative Study using Single Stage CNN Object Detectors," in *2019 5th IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, 2019, pp. 1–3.

[32] S. Manzoor, S. H. Joo, and T. Y. Kuc, "Comparison of Object Recognition Approaches using Traditional Machine Vision and Modern Deep Learning Techniques for Mobile Robot," in *2019 19th International Conference on Control, Automation and Systems (ICCAS)*, 2019, pp. 1316–1321.

[33] K. Manjari, M. Verma, and G. Singal, "CREATION: Computational constRained travEl aid for objecT detection in outdoor eNvironment," in *2019 15th International Conference on Signal Image Technology and Internet Based Systems (SITIS)*, 2019, pp. 247–254.

[34] I. Khokhlov *et al.*, "Tiny-YOLO object detection supplemented with geometrical data," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, 2020, pp. 1–5.

[35] S. Cass, "Nvidia Makes It Easy to Embed AI: The Jetson Nano packs a lot of machine-learning power into DIY projects-[Hands on]," *IEEE Spectr.*, vol. 57, no. 7, pp. 14–16, 2020.

[36] A. Lukman, A. Harjoko, and C. K. Yang, "MFCC Feature Classification from Culex and Aedes Aegypti Mosquitoes Noise Using Support Vector Machine," in *2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT)*, 2017, pp. 17–20.