

Y-Net: Insect Counting and Segmentation using Deep Learning on Embedded Devices

Amin Kargar
Tyndall National Institute
University College Cork
Walsh Scholar
Cork, Ireland
amin.kargar@tyndall.ie

Dimitrios Zorbas
School of Engineering & Digital
Sciences, Nazarbayev University
Astana, Kazakhstan
dimitrios.zorbas@nu.edu.kz

Michael Gaffney
Horticulture Development Department
Teagasc Ashtown Food Research
Centre
Dublin, Ireland
michael.gaffney@teagasc.ie

Brendan O'Flynn
Tyndall National Institute
University College Cork
Cork, Ireland
brendan.oflynn@tyndall.ie

Salvatore Tedesco
Tyndall National Institute
University College Cork
Cork, Ireland
salvatore.tedesco@tyndall.ie

Abstract— Insect pests can pose a serious threat to food production and agriculture in general and can cause substantial crop damage and economic losses. Monitoring insect pest populations is essential to control and mitigate these losses. Traditional monitoring methods are considered by growers and agronomists to be time-costly as well as labour-intensive tasks, which ultimately means that in times of high activity on farms it is a task which often is neglected. This study proposes an automated vision-based insect segmentation and counting approach through the use of deep learning (DL) models developed particularly for embedded systems. An image dataset for our target insect, *Halyomorpha halys*, was first created using images captured by our IoT-enabled image capture system deployed in a fruit orchard. Then, a Y-Net model inspired by U-Net was developed with the capability of insect counting in addition to segmentation. The performance of this model was assessed using a variety of different metrics, and the results demonstrated the feasibility and effectiveness of the model in counting and segmentation of insects using Edge-AI algorithms capable of running on embedded systems. Based on the achieved results, the proposed Y-Net model achieved a Mean Squared Error (MSE) of 1.9 for the insect counting task, an Intersection over Union (IoU) of 84.5% and a Dice Similarity Coefficient (DSC) of 91.5% for the segmentation task, with an inference time of nearly 0.4 seconds on a smartphone.

Keywords—Image segmentation, Object counting, Deep learning, CNN-based architecture, Insect monitoring, Precision agriculture.

I. INTRODUCTION

Insects have a significant impact, both positive and negative on different aspects of our lives despite their small size. Insect pests threaten our food security which is one of the primary basic needs for our society by reducing crop yields and crop quality. Annually, over 220 billion dollars in revenue are lost because of insect pests that destroy 40% of our crops [1]. A particularly damaging example of these insects is *Halyomorpha halys* (Hemiptera: Pentatomidae) (HH), which is an invasive shield bug native to east Asia, currently also existing in Europe causing €588 million losses to northern Italy orchards in 2019 [2], [3].

To reduce these losses, growers must visit and monitor their orchards regularly to recognize the presence of insect pest species and estimate the size of their populations. Then, this information is used to make decisions on the necessity of

pesticide usage to reduce these populations, a process that needs to be as frequent as possible. Therefore, early insect detection is critical since this improves the potential for good control of these pests and reduces the overall level of crop damage while also reducing the impact of pesticide usage on food quality and the environment [4]. This traditional style of insect monitoring typically requires a significant amount of time since growers must visit their orchards regularly, especially for large-scale orchards, and often needs a high level of taxonomical training to correctly identify insect species [5][6].

In recent years, with the progress of Information and Communication Technology (ICT), several vision-based methods that benefit from machine learning techniques have been developed by researchers to effectively monitor insects in orchards [7][8][9]. These systems are deployed in orchards and by analysing the images of orchards, they provide the required information about the presence and the population of insects in the field to growers. Regarding the image analysis aspect, Deep Learning (DL) models are contemporary methods that have attracted significant attention in recent years to automate insect detection and insect counting in orchards [10].

This study is part of a Horizon EU project named HALY.ID [11] whose aim is to monitor insects, specifically *Halyomorpha halys*, using innovative ICT tools. In this study, we proposed a new and accurate DL model to count and segment images containing the target insects, such as *Halyomorpha halys* (HH) in orchards. To achieve this aim, we first developed our insect image dataset collected using an IoT device [12] which was deployed in a pear orchard in Italy during the growing season in 2023. Then, an accurate CNN-based DL model was proposed for insect segmentation and counting. The proposed model, named Y-Net, was inspired by the U-Net model [13] that counts the HHs on the image in addition to segmentation. In contrast to U-Net which just performs the segmentation task, Y-Net has two separate outputs from distinct parts of the network to count and segment HH insects on the input image. The segmentation output provides an image that separates foreground (the HHs) from the background, and the counting output provides a number representing the number of HHs on the image. These main contributions are summarized as follows:

- Y-Net model inspired by the U-Net is proposed for insect counting in addition to insect segmentation in one stage.
- The model is a lightweight DL model suitable for edge-based systems running on embedded devices with restricted computational resources.

The remainder of this paper is organized into four sections as follows: In Section II, some related works are reviewed. Section III describes the device which was deployed in the orchard to collect the dataset. The collected dataset and its features are also presented in this section. In Section IV, the proposed CNN-based model used for counting and segmentation is also introduced. Section V describes the results obtained from the model that was developed for segmentation and counting. Finally, Section VI presents conclusions and recommendations for further research in this area.

II. RELATED WORK

In recent years, researchers have used DL techniques in insect monitoring systems to develop more accurate and efficient insect monitoring systems. This section illustrates some general and common DL models used for insect monitoring and also some studies that specifically work on the target insects of this study.

There are several previous studies that used object detection algorithms such as You Only Look Once (YOLO), regional-convolutional neural network (R-CNN), or Faster R-CNN for insect detection. For example, Mamdouh et al. [14] proposed a deep learning model which is a modified version of the YOLOv4 to detect and count olive fruit flies in the images captured by the smart pheromone traps. Their modified YOLOv4 obtained a precision of 84%, a recall of 97%, an F1-score of 90%, and a mean Average Precision (mAP) of 96.68%. Li et al. [15] proposed the TPest-RCNN model based on a Faster R-CNN to identify whiteflies and thrips in a greenhouse environment. They achieved an F1-score of 0.94 and a mAP of 0.95 for the detection of whiteflies and thrips using the proposed model. Similarly, Wang et al. [16] suggested a new architecture for apple pests recognition and counting based on Faster R-CNN named MPest-RCNN. The authors also used the ResNet101 feature extractor to improve the recognition precision achieving 99.1% and 99.5% for mAP and F1-score, respectively. Bereciartua-Perez et al. [5] proposed a two-stage approach based on deep learning to count whiteflies on eggplant leaves using mobile devices. The proposed model first segmented the leaf and removed the background and then density map estimation was used for insect counting. In this study, MAE of 3.36, RMSE of 7.84, and R2 of 0.97 were obtained using the proposed two-stage method.

In terms of our target insects, *Halyomorpha halys*, Dinca et al. [10] used YOLO models to detect the insects on images that were captured using a camera drone flying in an orchard. Two different versions of the YOLO, v5 and v8, were compared for performance and the results showed 94.6% accuracy for YOLOv8 and 90.9% for YOLOv5. In another study, Sorbelli et al. [17] proposed a YOLO-based model to detect this insect using drones in orchards. The author created a dataset captured using drones and other devices such as smartphones. They used the YOLO framework for insect detection and conducted a preliminary screening process on the dataset samples to improve the performance of the

detection model. Several YOLO models and metrics were evaluated, and the testing results demonstrated precision of over 89% and recall of 73%. In both studies, the dataset was created using images captured from plants in orchards infested by the insect. However, *Halyomorpha halys* has a brownish colour which makes it easily blend and hide with the background elements in orchards, such as tree branches and trunks. This significantly impacts the detection algorithm, thus a more complex model is needed to detect and extract the insect feature from the complicated background.

In this new study, we suggested using pheromone traps with a simple background to improve the image processing algorithm. This not only improves the detection accuracy but reduces the DL model complexity and size which is critical as we aim to implement the model on embedded devices with limited resources. Furthermore, instead of a general detection algorithm, such as YOLO, a lightweight segmentation-based model is proposed which not only segments the target insect on the image but also counts them.

III. MATERIALS AND DATASET

A. IoT Device

The device used to collect the dataset is shown in Fig. 1 and is described in detail in our previous work [12]. This device is equipped with a camera board to capture images from a double-sided trap. The camera board is an OpenMV Cam H7 Plus which is a low-power and small board based on a microcontroller (STM32H7) supporting Python scripting that makes deployment simple. The trap has a sticky surface that catches HHs with the help of a particular pheromone lure which only attracts HHs and consequently reduces bycatches. Moreover, a trap-based device was suggested to decrease the background complexity of the captured images. In fact, a white trap was chosen because it contrasts most with the brownish colour of the target insect facilitating the machine learning model training to distinguish the insects.

The device was adjusted to capture 2-megapixel images daily during the night using the provided LED. Timing the system to operate at night allows us to control lighting conditions and eliminate ambient lights and shadows; therefore, the captured images have the same conditions in terms of lighting parameters such as light intensity and brightness. In addition, to capture images from both sides of the trap, a servo motor was built into the device, enabling it to cover both sides of the trap for image capturing.

This device was deployed in a pear orchard in the Emilia Romagna region in northern Italy which was infested with *Halyomorpha halys* between 2022 and 2023.



Fig. 1. Deployed device in an orchard for dataset collection from a sticky double-sided trap.

B. Dataset

The most important first step in developing DL models for insect monitoring systems is to have an appropriate dataset of the target insect. In this study, we created a dataset for HHs using images captured by the device described previously. This first version of the dataset was collected during the farming season of 2023 and consists of over 240 images from the trap surface taken at night. These are 2-megapixel images with a resolution of 1600x1200. This is a binary segmentation dataset with two classes including objects and background, thus the mask images are black and white: the white areas represent objects (e.g., HH insects) and the black regions represent the background. The dataset consists of three folders:

1. Images: Contains trap images
2. Masks: Contains corresponding ground truth images that mask HHs
3. Num HHs: Contains a CSV file with the numbers of HHs on every image.

Fig. 2 shows some images of this dataset with ground truth masks and numbers of HHs. It should be acknowledged that despite using a specific pheromone which only attracts HHs, captured images still have lots of by-catches, such as flies or bees, which is also obvious in the sample images in Fig. 2.

In this study, for training and evaluating the proposed model, the dataset was split into training, validation, and test sets in which approximately 70%, 15%, and 15% of the total data were assigned to each set respectively. As mentioned before, images were taken from a two-sided trap, and the data analysis revealed that there is a difference between the distribution of insect numbers in images of these two sides. Therefore, to preserve this difference in all sets, stratified sampling based on the number of insects in images was used to split the data. To this purpose, images of each side were first split into train, validation and test sets separately using the stratified sampling technique and then combined to create the final dataset. This splitting strategy assures that the original dataset's insect number distribution is reflected in all the

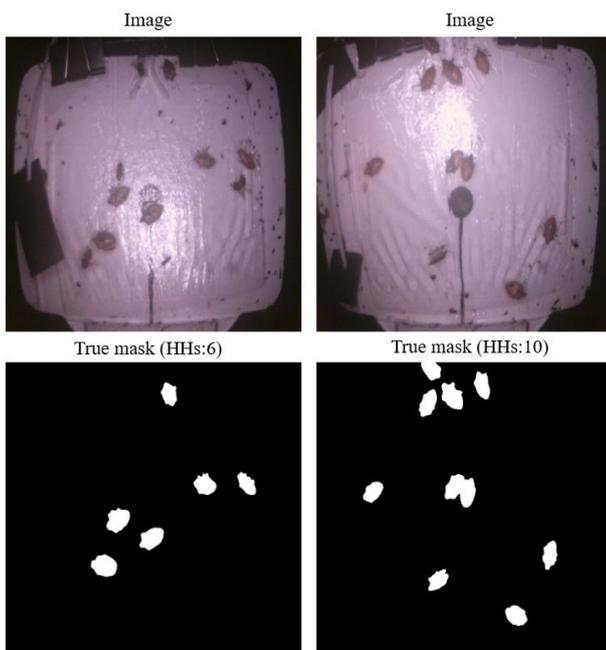


Fig. 2. Two samples of the dataset.

subsets. To this end, the following steps were followed to create train, validation, and test sets:

1. Divide images into side1 and side2 based on whether they were captured from side1 or side2.
2. Use stratified sampling to split the data for training, validation, and test sets for images from side1 (train_side1, validation_side1, test_side1).
3. Use stratified sampling to split the data into training, validation, and test sets for images from side2 (train_side2, validation_side2, test_side2).
4. Combine training sets, train_side1 and train_side2, to construct the final training set. The same was done to construct the validation and test sets.

Moreover, several augmentation techniques were employed on the training dataset to enhance the model generalization and decrease the risk of overfitting issues. By doing so, we added new variations of existing images to the training dataset and increased the number of samples. The augmentation techniques used were vertical and horizontal flip, rotation, and random brightness and contrast.

IV. Y-NET MODEL FOR INSECT COUNTING AND SEGMENTATION

Image segmentation is widely used to distinguish object regions in an image. One of the most popular image segmentation architectures is U-Net which is widely used for this purpose. This model has two main components, encoder and decoder, which are also known as the contracting and expansive paths respectively. The encoder part consists of convolutional layers that are widely used to extract important features from the input and decrease the dimensions of the input image. The decoder component is responsible for creating a segmentation map using upsampling layers and increasing the dimensions. The bottleneck is a central point where these two parts are connected to each other. In addition, U-Net benefits from skip connections which connect corresponding layers of encoder and decoder [13].

U-Net generates an output image with the same size as the input image highlighting the object regions present within the input image. However, for insect monitoring, the number of insects on the image is also required in addition to segmented areas. To count the objects from the segmented binary image generated by U-Net, methods such as Connected-component labeling (CCL) [18] method can be used to count the connected pixels in the binary image. But such approaches impose extra stages on the system and increase the algorithm processing time and consequently increase the power consumption which is a critical factor for edge-based systems running on embedded devices. Besides, using CCL may count objects incorrectly if they overlap; this could be problematic in case of a high overlap rate which occurred a lot in our dataset. To tackle these, here a new DL model to detect and count the target insect on trap images is proposed. To this purpose, the Y-Net architecture, inspired by U-Net, was proposed to count the target insects and segment them on the input image in one stage. This architecture is shown in Fig. 3. As shown in Fig. 3, a branch has been taken from the bottleneck of the U-Net model for the counting output. As a result, following the encoder part extracting features, the model presents two branches, one used for insect counting while the other one is used for the segmentation task.

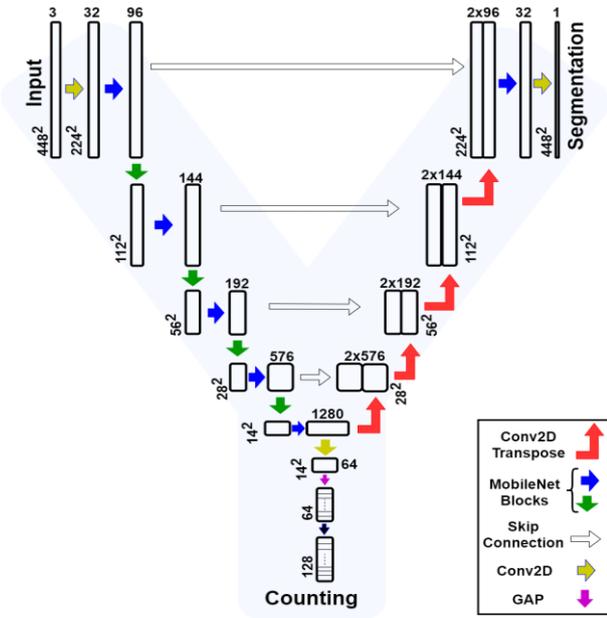


Fig. 3. Y-Net architecture with two outputs: counting and segmentation.

In this study, MobilenetV2 architecture [19] was used for the encoder component to efficiently extract features. MobilenetV2 was used due to its low complexity and lightweight structure which makes it well-suited for embedded devices with restricted computational resources.

The decoder part consists of several transposed convolution or deconvolution layers to upsample the feature maps and reconstruct high-resolution spatial information. Like U-Net, this Y-Net architecture benefits from skip connections that copy and concatenate encoder layers' information to the corresponding decoder layers. In the counting part, the global average pooling (GAP) layer was used instead of a fully connected layer to flatten the input feature maps into a vector. This vector is then fed to a dense layer with ReLU as an activation function, and eventually to an output layer with a linear activation function to predict the number of objects. This network was named Y-Net because of its Y-shaped structure.

It should be noted that using GAP instead of a fully connected layer significantly reduces the parameter numbers leading to a significant reduction in model size and computation; this is critical when the model is required to be run on embedded devices. Besides, this layer helps the network avoid overfitting issues since it averages all elements of each feature map while the fully connected layer flattens all elements of each feature map [20].

V. RESULTS EVALUATION AND DISCUSSION

A. Evaluation Metrics

As mentioned in Section IV, the proposed architecture has two outputs, counting and segmentation.

For the counting part, the Mean Squared Error (MSE) and Mean Absolute Error (MAE) were utilized for assessing the performance of the proposed model. MSE measures the mean squared difference between the number of predicted insects in the input image and the actual one. MAE evaluates the counting performance by computing the mean absolute difference between the predicted and actual number of insects. These errors are calculated as shown in Eq. (1)-(2):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2)$$

Where n is the number of images, and y_i and \hat{y}_i are the actual and predicted number of insects on each image, respectively.

For the segmentation part, four metrics, including Intersection over Union (IoU), Dice Similarity Coefficient (DSC), precision, and recall were used [21][22]. IoU measures the overlap ratio between the predicted segmented area to the ground truth mask, providing a segmentation task accuracy. The DSC is another metric commonly used for binary segmentation, this metric measures the similarity of the predicted segmented area and the ground true mask. Since the false positive and false negative factors are important when working on the number of insects, precision and recall are also used in this study. In this context, precision measures the ability of the model to avoid false negatives by computing the ratio of correctly classified positive pixels to all pixels predicted as positive. Additionally, recall measures the model's performance in correctly classifying all positive pixels, thus indicating the model's effectiveness in minimizing false negatives. These metrics are calculated as shown in Eq. (3-6):

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union} = \frac{TP}{TP+FP+FN} \quad (3)$$

$$DSC = \frac{2 \times TP}{2 \times TP + FP + FN} \quad (4)$$

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

Where TP is the true positive pixels, FP is the false positive pixels, and FN is the false negative pixels.

In addition to these metrics, we assessed the performance of the model in terms of inference time, which is also known as execution time or latency, and the number of parameters used in the model. These two metrics were categorized as efficiency metrics since they are related to the deployment and operation of the proposed model.

B. Implementation Parameters

The proposed Y-Net model was implemented and evaluated using TensorFlow on Google Colaboratory. For training the model, Binary Crossentropy was selected as the loss function for the segmentation task since we have two classes (HHs and background), and MSE was used for the counting task. Moreover, Adam optimization was utilized for model training during 100 epochs.

Furthermore, the performance of the proposed Y-Net was investigated for two different input sizes including 448x448 and 320x320 to compare the input size impact on the segmentation, counting and efficiency of the model. Also, we assessed the performance of the model by changing the alpha α parameter of MobilenetV2; alpha is a parameter defined to control the model width by scaling the channel number of each layer which impacts the model complexity and accuracy. Moreover, inference time of the model was measured by running the model on a Samsung A54 Android phone.

Table I. Y-NET PERFORMANCE ON THE TEST SET FOR TWO DIFFERENT INPUT SIZED AND FOUR DIFFERENT ALPHA VALUES OF MOBILENETV2 WITH MEAN VALUE AND 95% CONFIDENCE INTERVAL (CI) OF SEGMENTATION AND COUNTING METRICS.
IOU: INTERSECTION OVER UNION , DSC: DICE SIMILARITY COEFFICIENT, MSE: MEAN SQUARED ERROR, MAE:MEAN ABSOLUTE ERROR.

Model		Segmentation metrics				Counting metrics		Efficiency metrics	
Input Size	α	IoU	DSC	Precision	Recall	MSE	MAE	Model parameters	Inference time (s)
448x448	1	84.60 (84.39, 84.82)	91.56 (91.42, 91.70)	92.39 (91.97, 92.81)	91.65 (91.18, 92.12)	3.12 (2.64, 3.60)	1.34 (1.23, 1.46)	12,446,226	0.94
	0.75	84.01 (83.71, 84.30)	91.22 (91.03, 91.40)	91.90 (91.07, 92.72)	92.02 (90.86, 92.82)	2.38 (1.98, 2.78)	1.11 (1.00, 1.22)	8,897,226	0.75
	0.5	82.38 (82.07, 82.68)	90.18 (89.97, 90.39)	92.79 (92.42, 93.16)	89.21 (88.69, 89.74)	3.13 (2.59, 3.76)	1.20 (1.07, 1.36)	5,535,154	0.34
	0.35	81.21 (80.86, 81.56)	89.51 (89.27, 89.75)	93.97 (93.38, 94.47)	87.43 (86.41, 88.45)	2.58 (2.36, 2.88)	1.18 (1.09, 1.28)	3,842,194	0.31
320x320	1	84.96 (84.79, 85.10)	91.78 (91.68, 91.87)	91.90 (91.48, 92.31)	93.01 (92.68, 93.35)	2.89 (2.31, 3.47)	1.26 (1.15, 1.38)	12,446,226	0.48
	0.75	84.33 (84.14, 84.52)	91.43 (91.32, 91.54)	92.02 (91.37, 92.66)	92.77 (92.12, 93.43)	1.86 (1.65, 2.18)	0.98 (0.91, 1.08)	8,897,226	0.39
	0.5	82.73 (82.61, 82.84)	90.45 (90.37, 90.52)	92.34 (91.81, 92.87)	90.32 (89.61, 91.03)	3.14 (2.82, 3.45)	1.28 (1.20, 1.37)	5,535,154	0.18
	0.35	80.77 (80.56, 80.97)	89.22 (89.09, 89.35)	93.42 (93.16, 93.69)	87.91 (87.38, 88.45)	2.73 (2.41, 3.05)	1.22 (1.10, 1.34)	3,842,194	0.14

C. Results

Table I reports the values of these metrics for different configurations of the proposed Y-Net. It is evident that the model with the alpha value of 0.75 and input size of 320x320 had the lowest value of MSE and MAE at around 1.86 and 0.98, respectively. Based on the results, it was observed that the counting was affected when there was a multitude of HHs on the trap, specifically when the insects were laid over each other since it increased the complexity of extracting features of distinct insects for the model.

Based on Table I, the models with the input size of 448x448 ($\alpha=1$) and 320x320 ($\alpha=1$) have the best performance in terms of insect segmentation at approximately 85% for IoU and 91% for DSC. Additionally, these two models obtained a recall and a precision of about 92% indicating that the model is able to correctly predict positive cases which are actual positive and also minimise false positives and negatives. However, the model with an input size of 448x448 requires 0.94 seconds to analyse one image while the model with an input of 320x320 needs 0.48 seconds.

As expected, by decreasing the alpha, the model performance decreases for both counting and segmentation,

but it also results in a significant reduction in the number of model parameters and inference time. For example, by using a model with an alpha value of 0.35, IoU and DSC are respectively approximately 4% and 2% lower than a model with $\alpha=1$ but it has a model parameter and inference time of nearly three times less. There is therefore a trade-off between model efficiency and accuracy and should be carefully balanced for practical applications of automated insect monitoring on embedded devices.

Moreover, the number of parameters is independent of the input size, and this is attributed to the use of a GAP layer instead of a fully connected layer that made the model size and parameters unaffected by the input size. However, the input size has a direct impact on the model inference time. As mentioned, a model with an input size of 448x448 needs approximately twice as much time as a model with an input size of 320x320 for each prediction. Thus, a larger input size leads to higher execution time since a larger image needs more computations to be analysed by the model.

Finally, Fig. 4 depicts how the model (input size of 448x448 with $\alpha=1$) works by showing two samples of the model outputs, the first row is an image with three HHs and

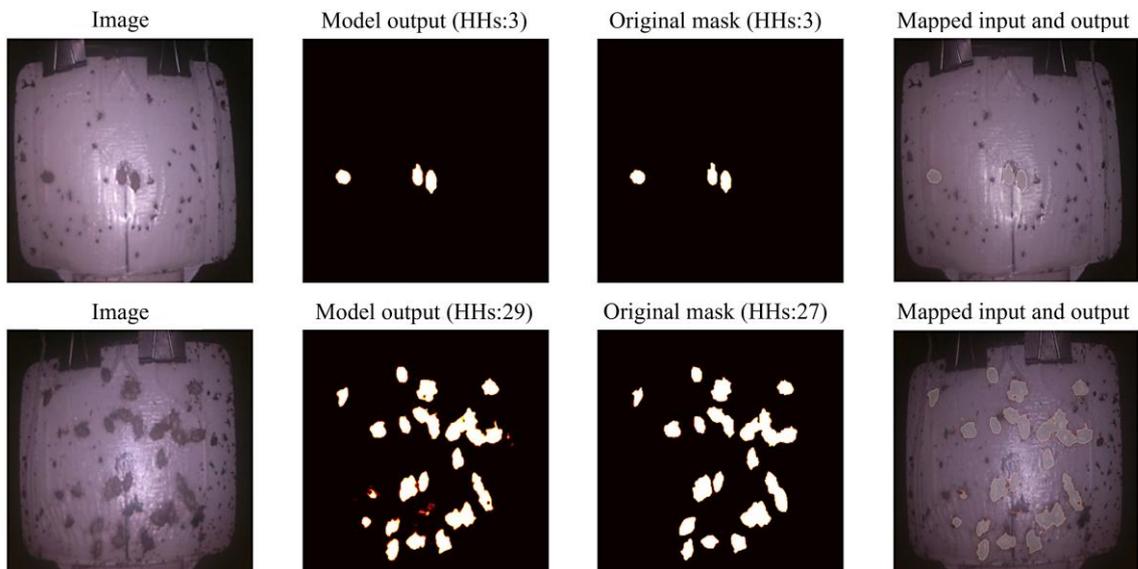


Fig. 4. Examples of the model outputs

the second row is an image with 27 HHs, some of which overlap. The first column shows the original image, the second column shows the model output including the segmented image and number of HHs, the third column shows the true mask and the actual number of insects on the input, and the last column visualizes segmentation results on the input image. As expected, the increase in the number of insects increases the chance of overlap among insects which negatively affects the performance.

VI. CONCLUSION

This study presents a deep learning-based model for insect counting and segmentation on images. The proposed model, named Y-Net, is inspired by the U-Net architecture, with the difference that in addition to segmentation, it also allows insect counting. This study focused on *Halyomorpha halys* and attempted to segment and count this species on images. The dataset was gathered using an IoT device deployed in an orchard which was infested by the targeted insect. Results obtained were promising and showed an MSE value of 1.9 in terms of insect counting, and in terms of segmentation the model obtained an IoU of 84.5%, a DSC of 91.5%, and precision and recall of over 92%.

In the future, we intend to implement the model on resource-constrained embedded devices on the edge enabling practical applications of automated insect monitoring by optimising the model, specifically in terms of energy consumption and RAM usage.

ACKNOWLEDGMENTS

This project is co-funded by the European Regional Development Fund (ERDF) under Ireland's European Structural and Investment Funds Programmes 2014–2020. This work was carried out as part of the Haly.ID project -- 2020EN508 funded by Ireland's Department of Agriculture, Food and the Marine under Grant: 2020 Trans National ERANET. The first author is supported by a Walsh Scholarship funded by Teagasc, The Irish Food and Agriculture Authority. Aspects of this work have been supported by Science Foundation Ireland under Grant 12/RC/ 2289-P2-INSIGHT, 13/RC/2077-CONNECT-P2, 16/RC/3835-VISTAMILK, and 16/RC/3918-CONFIRM. Moreover, we would like to express our gratitude to postdoctoral researcher, Dr. Mariusz Wilk for his contribution to the design and development of the IoT device.

REFERENCES

- [1] "FAO - Climate change fans spread of pests and threatens plants and crops, new FAO study." Accessed: Apr. 24, 2023. [Online]. Available: <https://www.fao.org/news/story/en/item/1402920/icode/>
- [2] V. Ferrari *et al.*, "Evaluation of the potential of near infrared hyperspectral imaging for monitoring the invasive brown marmorated stink bug," *Chemometrics and Intelligent Laboratory Systems*, vol. 234, p. 104751, Mar. 2023, doi: 10.1016/J.CHEMOLAB.2023.104751.
- [3] E. Costi, T. Haye, and L. Maistrello, "Biological parameters of the invasive brown marmorated stink bug, *Halyomorpha halys*, in southern Europe," *J Pest Sci (2004)*, vol. 90, no. 4, pp. 1059–1067, Sep. 2017, doi: 10.1007/S10340-017-0899-Z/TABLES/4.
- [4] E. C. Tetila, B. B. MacHado, G. V. Menezes, N. A. De Souza Belete, G. Astolfi, and H. Pistori, "A Deep-Learning Approach for Automatic Counting of Soybean Insect Pests," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 10, pp. 1837–1841, Oct. 2020, doi: 10.1109/LGRS.2019.2954735.
- [5] A. Bereciartua-Pérez, L. Gómez, A. Picón, R. Navarra-Mestre, C. Klukas, and T. Eggers, "Insect counting through deep learning-

- based density maps estimation," *Comput Electron Agric*, vol. 197, p. 106933, Jun. 2022, doi: 10.1016/J.COMPAG.2022.106933.
- [6] J. Sütő, "Embedded System-Based Sticky Paper Trap with Deep Learning-Based Insect-Counting Algorithm," *Electronics 2021, Vol. 10, Page 1754*, vol. 10, no. 15, p. 1754, Jul. 2021, doi: 10.3390/ELECTRONICS10151754.
- [7] D. J. A. Rustia *et al.*, "Edge-based wireless imaging system for continuous monitoring of insect pests in a remote outdoor mango orchard," *Comput Electron Agric*, vol. 211, p. 108019, Aug. 2023, doi: 10.1016/J.COMPAG.2023.108019.
- [8] C. J. Chen, Y. Y. Huang, Y. S. Li, C. Y. Chang, and Y. M. Huang, "An AIoT Based Smart Agricultural System for Pests Detection," *IEEE Access*, vol. 8, pp. 180750–180761, 2020, doi: 10.1109/ACCESS.2020.3024891.
- [9] D. A. Pham, A. D. Le, D. T. Pham, and H. B. Vo, "AlertTrap: On Designing An Edge-Computing Remote Insect Monitoring System," *Proceedings - 2021 8th NAFOSTED Conference on Information and Computer Science, NICS 2021*, pp. 323–328, 2021, doi: 10.1109/NICS54270.2021.9701558.
- [10] A. Dinca, D. Popescu, C. M. Pinotti, L. Ichim, L. Palazzetti, and N. Angelescu, "Halyomorpha Halys Detection in Orchard from UAV Images Using Convolutional Neural Networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14135 LNCS, pp. 315–326, 2023, doi: 10.1007/978-3-031-43078-7_26/FIGURES/14.
- [11] HALY.ID, "HALY.ID Project." Accessed: Jun. 14, 2022. [Online]. Available: <https://www.haly-id.eu/>
- [12] A. Kargar, M. P. Wilk, D. Zorbas, M. T. Gaffney, and B. Q'Flynn, "A Novel Resource-Constrained Insect Monitoring System based on Machine Vision with Edge AI," *5th IEEE International Image Processing, Applications and Systems Conference, IPAS 2022*, 2022, doi: 10.1109/IPAS55744.2022.10052895.
- [13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9351, pp. 234–241, 2015, doi: 10.1007/978-3-319-24574-4_28/COVER.
- [14] N. Mamdouh and A. Khattab, "YOLO-Based Deep Learning Framework for Olive Fruit Fly Detection and Counting", doi: 10.1109/ACCESS.2021.3088075.
- [15] W. Li, D. Wang, M. Li, Y. Gao, J. Wu, and X. Yang, "Field detection of tiny pests from sticky trap images using deep learning in agricultural greenhouse," *Comput Electron Agric*, vol. 183, pp. 168–1699, 2021, doi: 10.1016/j.compag.2021.106048.
- [16] T. Wang, L. Zhao, B. Li, X. Liu, W. Xu, and J. Li, "Recognition and counting of typical apple pests based on deep learning," 2022, doi: 10.1016/j.ecoinf.2022.101556.
- [17] F. Betti Sorbelli, L. Palazzetti, and C. M. Pinotti, "YOLO-based detection of *Halyomorpha halys* in orchards using RGB cameras and drones," *Comput Electron Agric*, vol. 213, p. 108228, Oct. 2023, doi: 10.1016/J.COMPAG.2023.108228.
- [18] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, and Y. Chao, "The connected-component labeling problem: A review of state-of-the-art algorithms," *Pattern Recognit*, vol. 70, pp. 25–43, Oct. 2017, doi: 10.1016/J.PATCOG.2017.04.018.
- [19] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, Jan. 2018, doi: 10.1109/CVPR.2018.00474.
- [20] M. Lin, Q. Chen, and S. Yan, "Network In Network," *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, Dec. 2013, Accessed: Nov. 28, 2023. [Online]. Available: <https://arxiv.org/abs/1312.4400v3>
- [21] A. Kirimtat and O. Krejcar, "A Guide and Mini-Review on the Performance Evaluation Metrics in Binary Segmentation of Magnetic Resonance Images," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13920 LNBI, pp. 428–440, 2023, doi: 10.1007/978-3-031-34960-7_30/FIGURES/10.
- [22] D. Müller, I. Soto-Rey, and F. Kramer, "Towards a guideline for evaluation metrics in medical image segmentation," *BMC Res Notes*, vol. 15, no. 1, pp. 1–8, Dec. 2022, doi: 10.1186/S13104-022-06096-Y/FIGURES/2.