



Detection and Quantification of Glandular Trichomes (Bulbous) on Potato Plant Leaf Images Using Deep Learning

M. Fauzan Azhari ^{a,1}, Rohmatul Fajriyah ^{b,2,*}, Izzati Muhimmah ^{a,3}, Dan Jeric Arcega Rustia ^{c,4}, Marinus J.M. Smulders ^{d,5}, Micha Gracianna Devi ^{d,6}

^a Department of Informatics Master Program, Faculty of Industrial Technology, Universitas Islam Indonesia, Jl. Kaliurang KM 14.5, Yogyakarta 55584, Indonesia

^b Department of Statistics, Faculty of Mathematics and Natural Sciences, Universitas Islam Indonesia, Jl. Kaliurang KM 14.5, Yogyakarta 55584, Indonesia

^c Greenhouse Horticulture and Flower Bulbs, Wageningen Plant Research, Wageningen University & Research, 6708 PB Wageningen, The Netherlands

^d Plant Breeding, Wageningen Plant Research, Wageningen University & Research, 6708 PB Wageningen, The Netherlands

¹ m.fauzan.azhari@students.uui.ac.id; ² rfajriyah@uui.ac.id*; ³ izzati@uui.ac.id; ⁴ dan.rustia@wur.nl; ⁵ rene.smulders@wur.nl;

⁶ micha.devi@wur.nl

* Corresponding author

ARTICLE INFO

ABSTRACT

Keywords

glandular trichomes
potato
deep learning
object detection
YOLOv8

Potato plants have a very high nutritional value, making them widely cultivated in Indonesia. To ensure the cultivation of potatoes has good quality, many individuals, ranging from farmers to researchers and plant breeders, strive to explore and understand the characteristics of plant resistance sources, one of which is through the role of trichomes. Trichomes are fine hairs that coat the outer surface of plant leaves, serving as a physical barrier and regulating plant temperature. Identification and quantification of trichomes are commonly conducted manually by researchers, which consumes much time and is inefficient. Therefore, a system that can automatically detect and quantify trichomes is crucial to avoid manual identification and quantification, allowing these processes to be carried out more quickly. This study utilized a deep learning approach to train a model capable of detecting and quantifying trichome objects. The model architecture used was YOLOv8. From the training process, the resulting mean average precision (mAP) at a confidence threshold of 50 was 0.816, while the mAP at a confidence threshold of 90 was 0.38. This model is expected to assist experts or researchers in the field of agriculture in identifying trichomes, thereby optimizing crop yields.

1. Introduction

Indonesia heavily relies on the agricultural sector, which plays a crucial role in the welfare of its people. As a result, horticultural crops are widely cultivated, including potato plants. Potatoes are a significant source of carbohydrates and have excellent nutritional value, making them one of the most cultivated crops not only in Indonesia but also in European countries. According to data from the Central Statistics Agency in 2022, potatoes ranked fourth as the most produced and cultivated crops in Indonesia, with a total of 1.36 million tons in 2022.

To ensure that potato cultivation and production are of high quality, farmers, researchers, and plant breeders are striving to explore and understand the characteristics of new resistance sources, one of which is the role of trichomes as a defense mechanism in plants. Trichomes are fine hair-like structures that cover the outer parts of plant leaves and function as physical and chemical barriers [1], [2]. Trichomes play a crucial role as a plant defense mechanism against extreme weather conditions or insect pests, prompting researchers and plant breeders to identify trichome glands to produce high-quality plants.

Identification and quantification of trichomes are usually done manually by researchers. Manual techniques for trichome detection and quantification, while feasible for small datasets, are highly time-consuming and prone to human error. Given the complex morphology of trichomes and their often-subtle visual differences from other plant structures, manual observation can lead to inconsistencies in counting and categorization. Furthermore, as the scale of analysis increases, such as in high-throughput screening or large agricultural studies, the inefficiency of manual methods becomes more apparent [3], [4]. Deep learning addresses these challenges by automating the detection process, significantly reducing the time required and minimizing the potential for human error. Deep learning ability to process images quickly while maintaining high accuracy ensures that trichomes are consistently detected and quantified, providing reliable data even in large-scale studies.

Several studies have successfully applied deep learning in agricultural contexts. For example, the study titled 'Deep Learning Image Segmentation and Extraction of Blueberry Fruit Traits Associated with Harvestability and Yield' utilized deep learning for the digital phenotyping of blueberry fruits [5]. This study focused on developing a data processing pipeline capable of counting the number of berries, measuring ripeness levels, and evaluating crop yields automatically through image segmentation methods and deep learning algorithms to classify four blueberry cultivars. This research highlights the significant potential of deep learning in automating tasks that previously relied on manual observation, further supporting the validity of using similar technology for trichome detection in plant leaves [5], [6].

Understanding this problem, it is essential to develop a system or model based on deep learning that can automatically identify and quantify trichomes. This model is expected to assist experts or researchers in the agricultural field in identifying trichomes, thereby optimizing crop yields [7]. The method used to build the model is one of the deep learning algorithms, namely convolutional neural networks (CNN), with the YOLOv8 model architecture. You Only Look Once (YOLO) is one of the variant architectures in the CNN algorithm [8][9]. YOLO can perform real-time object detection by including localization, where the localization generated by YOLO is a collection of pixels with the highest-class probability. The localization process is carried out by applying the YOLO model to the image at each location (pixel) and then assigning a probability value to the image as a result of detection. The YOLO architecture uses a slightly different approach compared to other CNN variant architectures, where YOLO uses a single neural network across the entire image and partitions the image into regions, predicting bounding boxes and their probability values [10], [11].

Based on the background and problems mentioned, the author will attempt to develop a deep learning model that can be used to identify and quantify trichomes on the surface of plant leaves using one of the CNN variant architectures, YOLOv8, utilizing deep learning and computer vision technology. The research conducted in this study is expected to assist researchers, plant breeders, and plant scientists in automatically identifying trichomes.

2. Method

This research aims to train a model to detect and quantify trichome images on potato plant leaves using the CNN method, specifically employing the YOLOv8 architecture. The data used in this study consisted of microscopic images of the surface of cultivated potato leaves obtained from Wageningen University & Research.

The collected data underwent a preprocessing stage aimed at dividing the images into several sections (subimages) with dimensions of 4×4 . Trichome detection was performed through the

subimages to facilitate the system in identifying trichomes by enlarging the image size according to each subimage. An illustration of the image preprocessing can be seen in Fig. 1.

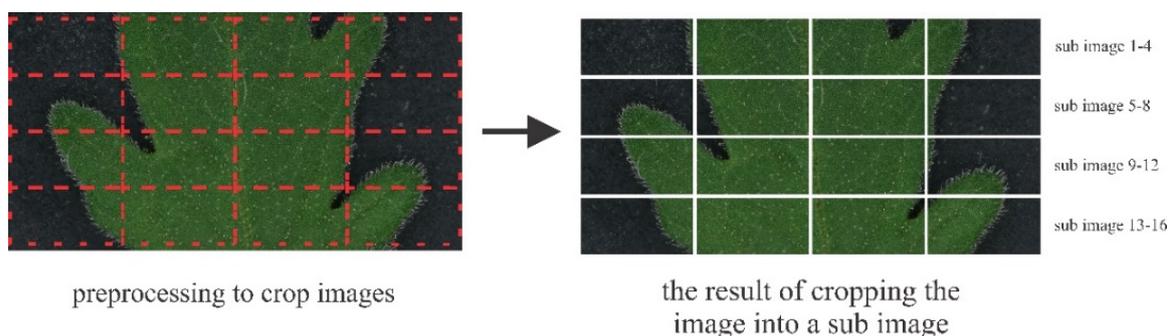


Fig. 1 Image preprocessing workflow for potato leaf trichome detection.

The next preprocessing step was applying data augmentation techniques such as image rotation, flipping, and scaling to artificially increase the diversity of the training dataset and prevent overfitting. This ensures the model generalizes well to unseen data. Additionally, the images were normalized to standardize pixel intensity values, making them more suitable for deep learning models. Normalization ensures that each pixel value is within a similar range, typically between 0 and 1, which helps improve model convergence during training.

The next step was the labeling process, assigning class information to the objects to be detected. This study focused on the class of glandular bulbous trichomes. These trichomes have short stalks that swell at the top, giving them a rounded or bulbous appearance. The next step involved configuring the system and tuning parameters for the neural network process. This process is crucial as it will affect the final results and the accuracy of the model in performing detection. The loss functions used in the YOLOv8 architecture include box loss, classification loss, and distribution focal loss. The metrics used were mean Average Precision (mAP) and Intersection Over Union (IoU). mAP was chosen because it provides a comprehensive measure of the model's performance by evaluating both precision and recall across different detection thresholds. Unlike simpler metrics like accuracy, which might not account for false positives and false negatives in object detection tasks, mAP gives a more nuanced insight into the balance between these two factors, allowing us to better understand how well the model is detecting trichomes. mAP was used to ensure that the model's ability to correctly detect glandular bulbous trichomes was thoroughly evaluated. Its metric has a value range between 0 and 1. This range is used to measure the quality of the object detection system's performance in object recognition tasks. The mAP value is obtained by calculating the Average Precision (AP) for each detected object class, and then averaging all those APs. AP itself measures the system's accuracy and precision in detecting objects by considering recall (the number of objects successfully detected) and precision (the number of correct detections). A higher mAP value indicates better performance, with 1 as the maximum value reflecting perfect detection. Meanwhile, a lower mAP value indicates less accurate performance or poor detection. The choice of threshold for determining whether an object detection is correct or incorrect can affect the mAP value. Common IoU threshold values are around 0.5 to 0.7, but they can be adjusted based on the needs of the research. With a value range of 0 to 1, mAP provides a comprehensive measure of object detection performance across different classes and facilitates comparison between different systems. In this study, a threshold value of 0.5 (50%) and 0.5-0.9 (50%-90%) was used as a reference for the success of the trained model.

After the learning model process was completed, evaluation or testing was carried out using validation data. This was also done to assess the implementation and final results of the trained model. To generalize the results obtained regarding trichome detection and quantification, several steps were done, such as comparing the number of trichomes predicted by the model with the actual number of trichomes in the ground truth data. Then, the precision and recall values based on the obtained true positive, false positive, and false negative values were calculated.

The output of this study included the detection and quantification of glandular bulbous trichomes on microscopic images of cultivated potato leaf samples. Quantification was conducted based on the partitioned areas or subimages. This model serves as one of the initial steps or pioneers in trichome detection and quantification using deep learning. Future research can develop models capable of detecting and quantifying other types of trichomes and apply them to leaf samples from other plant species. The overall research flow is illustrated in the flowchart shown in Figure 2.

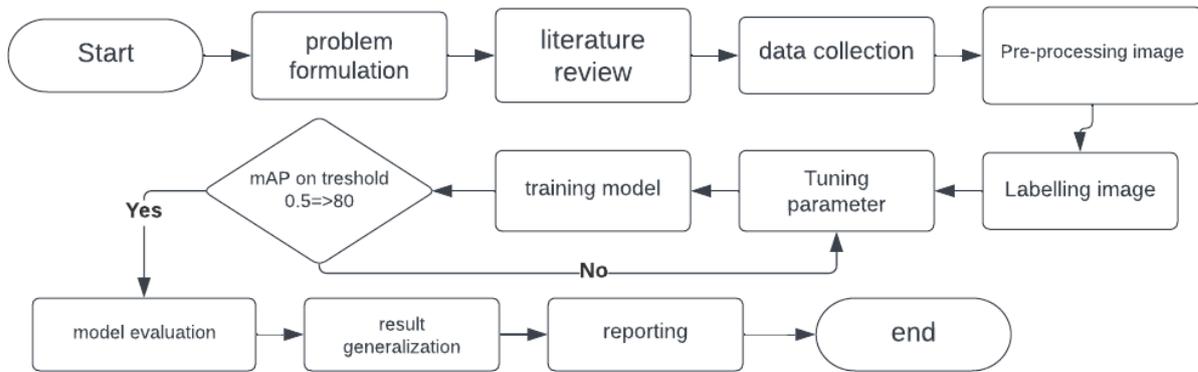


Fig. 2 Flowchart of the deep learning model training and detection process.

2.1. Convolutional Neural Network (CNN)

CNN is a type of neural network commonly used in image processing [12]. It performs feature learning during data preprocessing, which transforms image data into numerical features from pixel values for computer processing. This process involves several layers, including the convolution layer, pooling layer, and fully connected layer [13], [14]. After feature learning, pixel values are converted into vectors for input into the neural network algorithm. Thus, CNN consists of two main parts: feature extraction (preprocessing) and neural network (classification) [15].

Each layer in CNN has the following functions:

- Convolutional Layer: Performs convolution between filters (kernels) and the input image to extract basic features such as edges, textures, or specific patterns from the image.
- ReLU (Rectified Linear Unit) Layer: Applies an activation function that introduces non-linearity into the model by converting negative values to zero, helping the model capture non-linear relationships.
- Pooling Layer: Reduces the spatial dimensions of the feature maps (typically using max pooling or average pooling), decreasing the number of parameters and computations in the network, and making feature detection more robust to slight translations.
- Fully Connected Layer: Connects all neurons from one layer to neurons in the next layer, combining the extracted features to make the final decision, such as classifying objects within the image.
- Softmax Layer: Produces output probabilities for each class, ensuring the total probability sums to 1, and determines which class is the most likely outcome.

Each layer contributes to learning feature representations from the input image, with the features becoming more abstract at each successive layer [16].

2.2. You Only Look Once (YOLO)

The YOLO algorithm uses CNN for object detection [17]. As its name suggests, the YOLO algorithm uses only one layer of the neural network on the image [18]. This network divides the image into several regions and predicts bounding boxes and probabilities for each region simultaneously [19]. To calculate the confidence value of a bounding box, the following equation can be used:

$$confident = P_r(object) \times IOU_{pred}^{truth} \quad (1)$$

where $P_r(object)$ is the probability of an object being detected within the bounding box, and IOU_{pred}^{truth} is the ratio of the overlapping area between the predicted bounding box and the ground truth bounding box to the combined area of both bounding boxes. Equations (2) and (3) below show how YOLO calculates these probabilities [20].

$$P_r(Class_i|Object) \times P_r(Object) \times IOU_{pred}^{truth} = P_r(Class_i) \times IOU_{pred}^{truth} \quad (2)$$

$$Tensor\ Size = S \times S \times (B \times 5 + C) \quad (3)$$

where $P_r(Class_i|Object)$ is the conditional probability that the detected object belongs to class i , given the presence of an object in the bounding box, and C is the number of object classes the model aims to detect [21].

3. Result and Discussion

3.1. Data Preprocessing

The total number of images used for this research was 79. These 79 images underwent a preprocessing stage, where the images were cropped to a dimension of 4×4 , resulting in each image being divided into 16 new subimages. However, not all subimages generated were used during the labeling (annotation) process. Fig. 3 illustrates how the subimage filtering process is conducted to select which subimages will proceed to the annotation process. Subimages that did not represent the leaf surface or did not contain trichomes were removed and were not used in the labeling and training process. As a result, the total data produced from this process was 937 subimages.

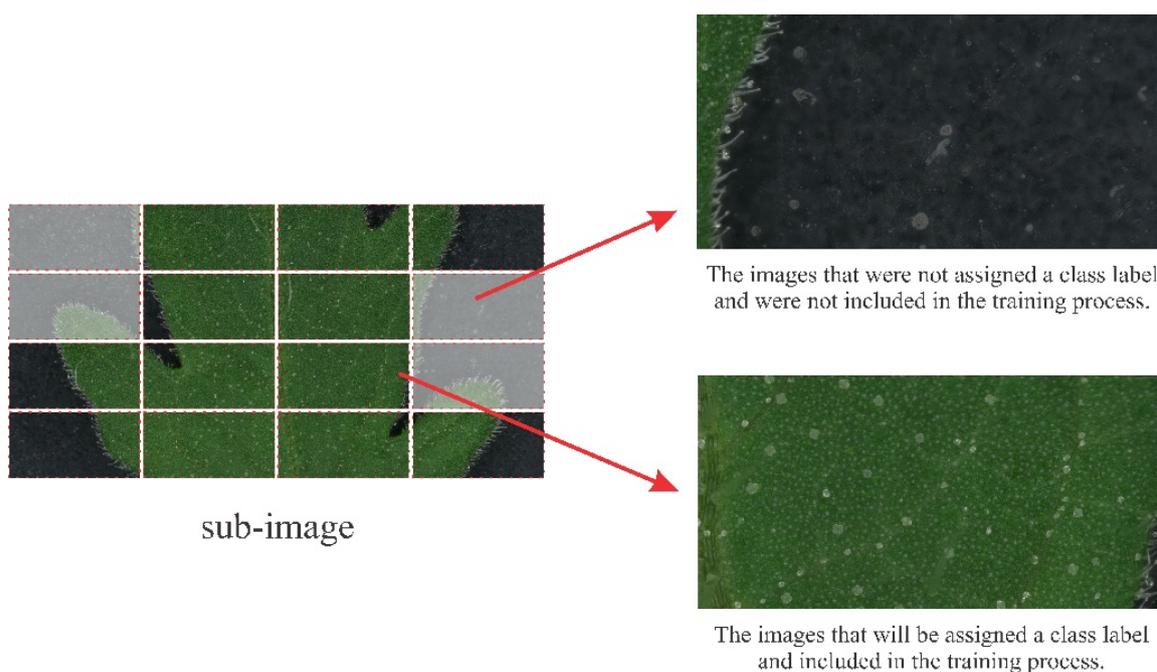


Fig. 3 Example of cropping and filtering subimages for training.

3.2. Dataset Splitting and Labelling

From the whole image that had been obtained, the data was divided into two parts: 80% for training the model, consisting of 750 images, and 20% for testing, consisting of 187 images. The training data was used to train the model, while the test data was used to measure the accuracy of the trained model.

After the dataset was divided, the next step was labeling each data entry. This process aimed to provide class annotations and place bounding boxes around each object in the specified class. All labeled results were exported into YOLO format (txt files). Each .txt file contains information about the object class code, object coordinates (x, y), width, and height. Fig. 4 shows an example of a data

sample resulting from annotation in YOLO format. As seen in the figure, it consists of 6 annotated objects, and each object contains information about the object class (0), object coordinates (x, y), object width, and object height.

object 1 ←	0	0.283899	0.203767	0.038994	0.074996
object 2 ←	0	0.630380	0.217311	0.034286	0.066298
object 3 ←	0	0.828221	0.223757	0.031506	0.064457
object 4 ←	0	0.851851	0.849908	0.030580	0.068140
object 5 ←	0	0.543275	0.406077	0.030580	0.060773
object 6 ←	0	0.436709	0.438306	0.028726	0.051565
	↓	↓	↓	↓	↓
	class	x	y	width	height

Fig. 4 Sample of annotated data in YOLO format with bounding boxes.

3.3. Training model

The initial step taken before starting the model training process was to set the hyperparameters. Hyperparameters in the context of machine learning and deep learning are parameters whose values are determined before the training process begins. Hyperparameters control certain aspects of the training process and model architecture that cannot be learned from the data during training. In this study, several important parameters such as epoch, learning rate, batch size, image size, and others were adjusted. These parameters are presented in more detail in Table 1.

Table 1. Hyperparameter Tuning for YOLOv8 in Trichome Detection Model Training

Parameter	Description	Value
epoch	A complete cycle where the entire dataset is processed by the learning algorithm to update the model weights.	100
Learning rate	Determining the magnitude of changes or adjustments made to the model weights during the training process each time the backpropagation algorithm updates the weights.	0.01
Batch size	The number of data samples processed together before the model updates its weights during training.	16
Patience	The number of epochs to wait when there is no improvement in validation metrics to prevent overfitting.	100
Imgsz	The target size for image data during the training process.	640
Single_cls	Combining all classes/labels in the dataset into a single class during training. This is useful for binary classification tasks or when focusing on object detection rather than classification.	True
Weight_decay	The parameter that controls the level of penalty on large weights to reduce overfitting.	0.0005
box	The size and scale of bounding boxes used for object prediction.	7.5
nms	Removing overlapping predictions with low confidence scores.	False

The following is a visual of the train_batch from several sample data during the training process, which has been conducted for 100 epochs. This visualization can help researchers check how the model learns and detects objects in the training images. It includes images with bounding boxes and the classes predicted by the model, allowing researchers to see whether the model detects objects correctly and in the right locations.

From the visualizations in Fig. 5, it can be observed that the detection of bounding boxes appears accurate and becomes increasingly precise as the training epochs progress. This indicates that the model is learning effectively.

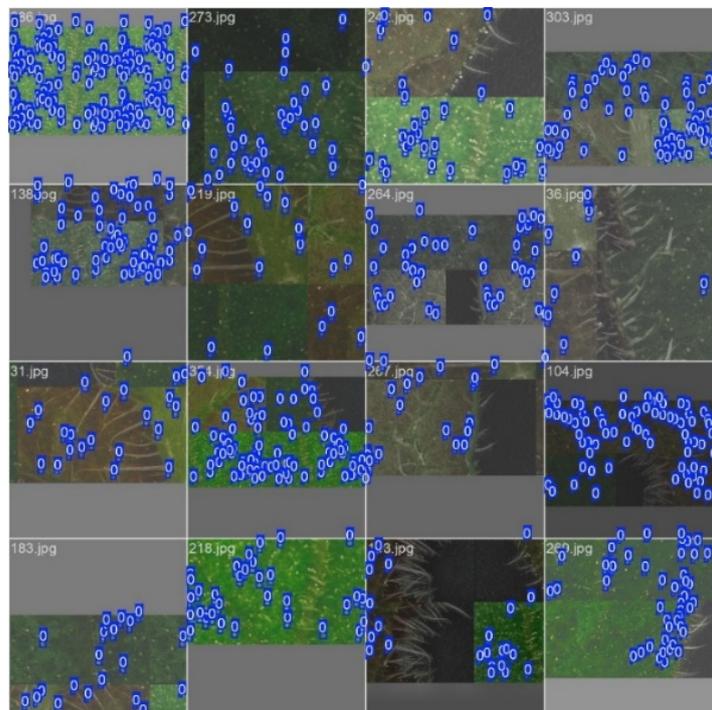


Fig. 5 Visualization of bounding boxes during the training process.

3.4. Evaluation Metrics

Once the training process is complete, the model will display several metrics or visualization plots that can be used to evaluate whether the training process has produced a good model or not. Some of the evaluation metrics generated include correlogram, box loss, classification loss, DFL loss, precision, recall, and mAP.

A correlogram is a visualization that shows the correlation relationships between several variables. In the context of machine learning and data analysis, correlograms are used to depict the degree and direction of correlation between pairs of variables, which can help identify patterns and relationships within the data. The correlogram displayed consists of four elements: x , y , width, and height. These elements relate to the size and location of the bounding box.

The visualization in Fig. 6 shows a fairly strong positive correlation between the height and width of the bounding box. This means that as the width of the bounding box increases, its height also increases. This is because the glandular bulbous trichome objects have a consistent shape or proportion as their size changes. The coordinates of the bounding box center (x , y) do not exhibit a strong correlation, meaning that the horizontal (x) and vertical (y) positions of the bounding box tend not to move together or are randomly distributed. This could be because the glandular bulbous trichome objects in the image do not move diagonally or follow any specific pattern.

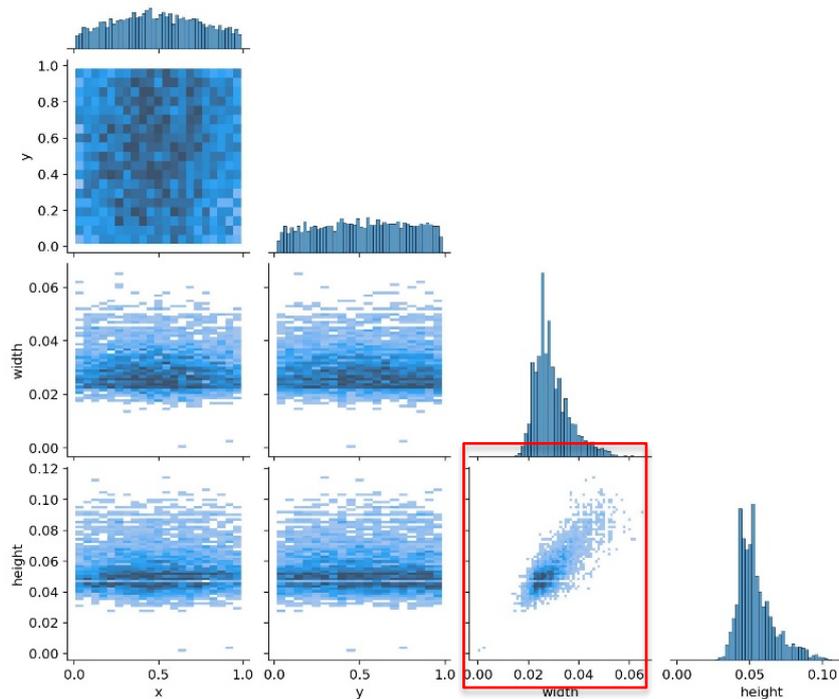


Fig. 6. Correlogram of bounding box features.

The visualization in Fig. 7 shows that each loss metric decreases significantly during the first 10-30 epochs and does not show a significant decrease after the 30th epoch, although there is still a gradual and consistent decline. For the precision, recall, and mAP metrics, they also experienced a significant increase during the first 30 epochs and continued to improve, despite some minor fluctuations, until the end of the training process. This indicates that the training process generally went well and produced a model with fairly good performance.

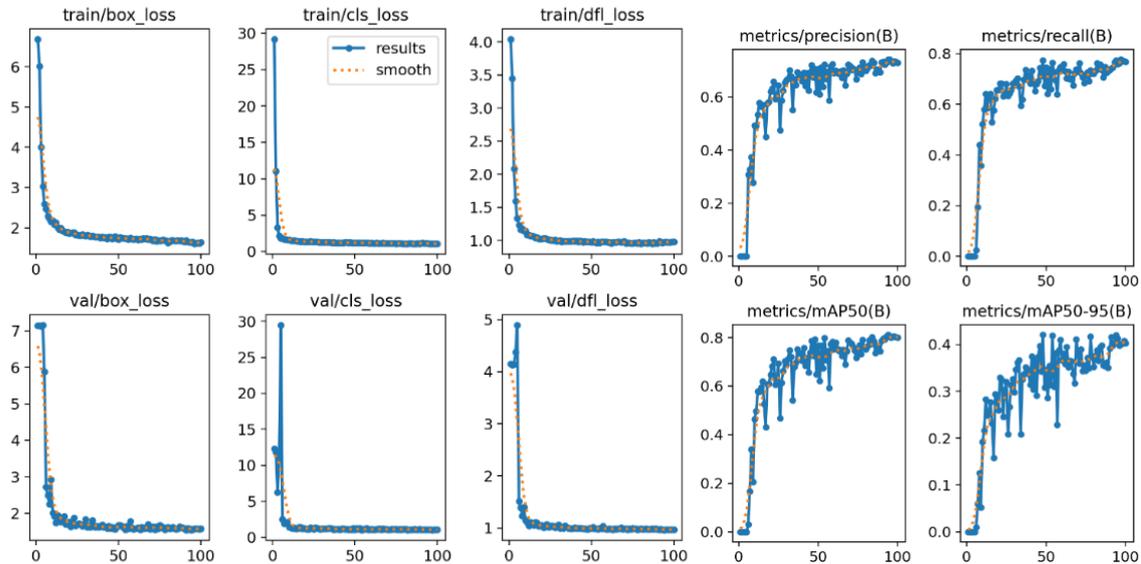


Fig. 7. Plot of loss functions, precision, recall, and mAP during training.

From the precision-confidence plot in Fig. 8, it can be concluded that the model achieved maximum precision (1.00) at a confidence threshold of 0.813, meaning that all positive predictions at this threshold are correct, with no errors. Additionally, the precision-recall plot shows that the mAP value at a confidence threshold of 0.5 was 0.812, indicating good model performance.

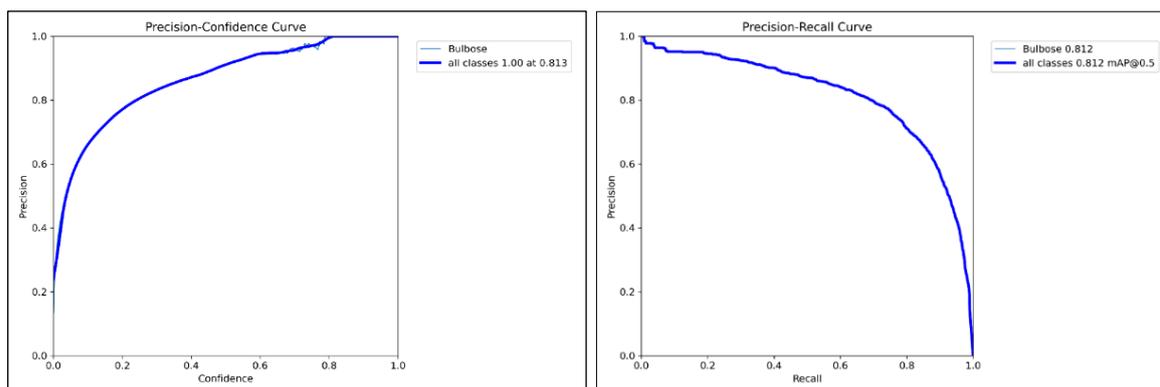


Fig. 8 Precision-confidence and precision-recall curves for the trichome detection model.

3.5. Model Implementation

Model testing was conducted to observe the implementation of the YOLOv8 model in detecting trichomes. Fig. 9 shows a visual representation of the model's implementation on one of the subimages in the validation data. In Fig. 9, the trained model performs well in detecting glandularis bulbous trichomes, with precise bounding box placement. During the detection process, a function was added to quantify the number of detected objects and display the results in the upper-left corner of the image. The quantification result shows that the model detected 24 trichomes in the subimage.

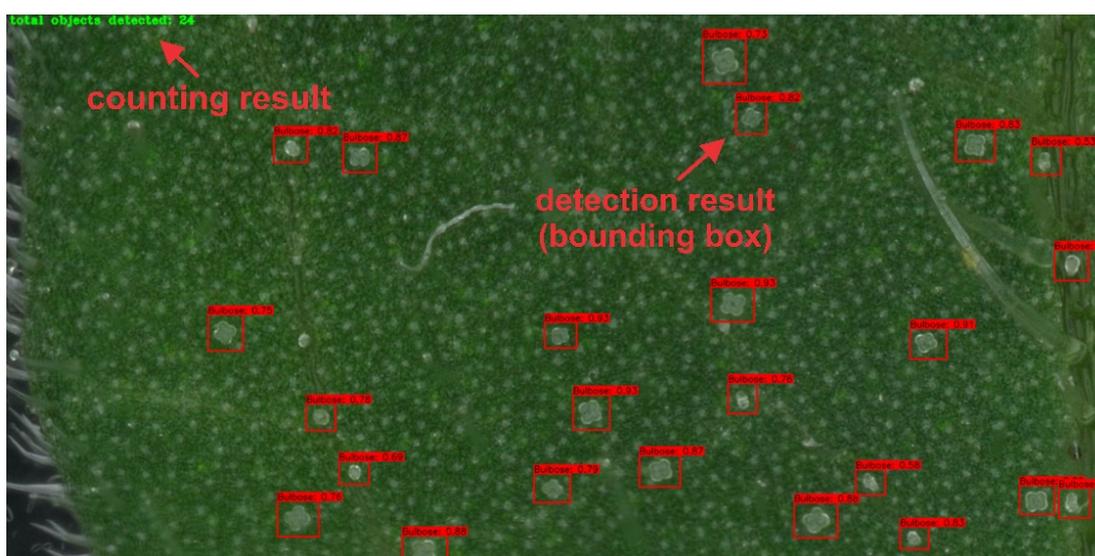


Fig. 9 Example of successful detection and quantification of trichomes by YOLOv8.

Fig. 10 present several examples where the YOLOv8 model failed to accurately detect trichomes. These cases provide insights into the model's limitations under certain conditions, which are important to evaluate in practical applications. Fig. 10(a) shows a situation where some trichomes were not detected by the model. This is likely due to the trichomes' shape and color appearing faint and resembling the surface of the leaf, making them difficult for the model to recognize. Smaller trichomes or those located in areas with low contrast are often not detected by the model. Additionally, Fig. 10(b) illustrates a situation where an object that is not a trichome was incorrectly detected as a trichome by the model. For example, dust particles or fine hairs on the surface of the leaf were interpreted as trichomes by the model due to their similar shape and visual texture. This error is likely caused by pattern similarities between trichomes and other small objects with similar contours or shapes, misleading the model in its classification. This case highlights the importance of a more diversified dataset to help the model better distinguish trichomes from visually similar but irrelevant objects.

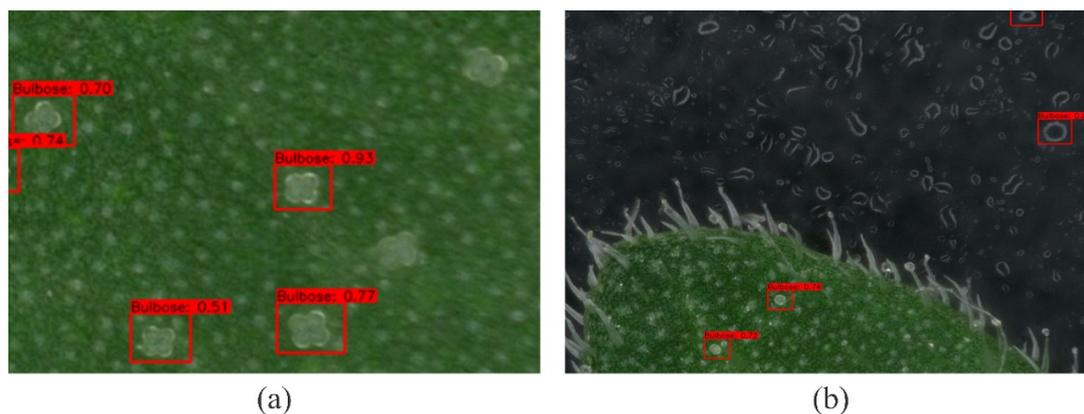


Fig. 10 Cases where YOLOv8 Model failed to detect trichomes accurately.

For the testing on 1 image of the entire leaf surface (all subimages), the results are shown in the Fig. 11. In Fig. 11, it can be seen that the model is able to detect trichomes across all sub-images very well, although there are some trichome objects that the model failed to detect. Additionally, there are a few errors where objects outside the leaf surface, which should not be trichomes, were predicted as trichomes by the model.

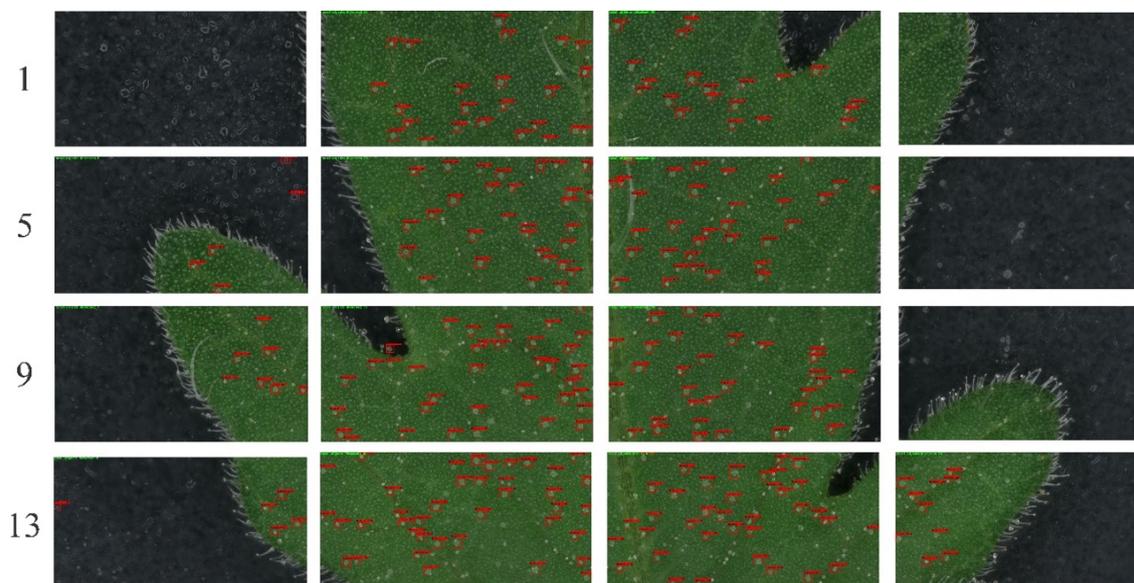


Fig. 11 Detection of glandular trichomes across multiple subimages of a leaf surface.

The details of the detection counts for each subimage are presented in Table 2. There are 3 variables that were calculated, namely “detected trichomes” (true positives), “undetected trichomes” (false negatives) and other objects incorrectly predicted as trichomes (false positives).

Table 2. Evaluation of Trichome Detection for Each Sub-Image of a Potato Leaf

Sub - Image	Detected Trichomes	Undetected Trichomes	Other objects incorrectly predicted as trichomes
Sub-Image 1	0	0	0
Sub-Image 2	24	2	0
Sub- Image 3	20	3	0
Sub- Image 4	0	0	0
Sub- Image 5	3	1	2
Sub- Image 6	30	3	0
Sub- Image 7	33	8	0

Sub - Image	Detected Trichomes	Undetected Trichomes	Other objects incorrectly predicted as trichomes
Sub- Image 8	0	0	0
Sub- Image 9	7	4	0
Sub- Image 10	37	4	0
Sub- Image 11	34	4	0
Sub- Image 12	0	1	0
Sub- Image 13	4	2	1
Sub- Image 14	39	8	0
Sub- Image 15	37	8	0
Sub- Image 16	10	5	0
Total	278	53	3

The results from the testing are then applied to (4) to calculate the total accumulated number of trichomes that predicted by model across all sub-images.

$$N = \sum_{i=1}^{16} T_i \tag{4}$$

$$N = \sum_{i=1}^{16} T_i$$

$$N = 0 + 24 + 20 + 0 + 5 + 30 + 33 + 0 + 7 + 37 + 34 + 0 + 5 + 39 + 37 + 10$$

$$N = 281$$

From (4), the total number of trichomes predicted by the model across all subimages is 281 trichomes. From the above equation, the total number of trichomes predicted by the model across all subimages is 281. Details of the number of valid data and the number of incorrectly predicted data will be presented in Table 3.

Table 3. Number of True Positive, False Positive, and False Negative Trichome Detections

Prediction	Value
True Positive (TP)	278
False Positive (FP)	3
False Negative (FN)	53

From Table 3, which details the number of predictions for true positive, false negative, and false positive cases, the precision and recall values can be calculated using the following equations:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \times 100\%$$

$$\text{Precision} = \frac{278}{278 + 3} \times 100\%$$

$$\text{Precision} = \frac{278}{281} \times 100\%$$

$$\text{Precision} = 0.98 \times 100\%$$

$$\text{Precision} = \mathbf{98\%}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \times 100\%$$

$$\text{Recall} = \frac{278}{278 + 53} \times 100\%$$

$$\text{Recall} = \frac{278}{331} \times 100\%$$

$$\text{Recall} = 0.839 \times 100\%$$

Recall = 83.9%

From the calculated precision, 98% of the images predicted to have trichomes contained trichomes, while 2% did not. The recall value indicates that the model successfully detected trichomes in 83.9% of the images where trichomes were truly present.

As a comparison, manual detection techniques performed by human researchers, while highly accurate under certain conditions, tend to be time-consuming and prone to errors due to fatigue or observer subjectivity. In terms of speed and efficiency, YOLOv8 demonstrates a significant advantage. Previously, other deep learning approaches such as Mask R-CNN and Faster R-CNN have been applied for object detection tasks in various contexts. However, YOLOv8 offers an edge in terms of real-time detection speed without sacrificing too much accuracy, making it more suitable for real-time applications like large-scale plant monitoring. That said, it is important to note that models like Mask R-CNN have an advantage in terms of more precise object segmentation, particularly for detecting smaller or hidden trichomes. Therefore, future research could explore hybrid methods or combine multiple deep learning approaches to achieve more optimal results.

4. Conclusion

The implementation of the YOLOv8 model for detecting and quantifying glandular bulbous type trichomes on cultivated potato leaf images was performed quite effectively. The test results using one image, which was partitioned into several sub-images, yielded a precision of 98% and a recall of 83.9%. Among the 16 sub-images tested, there were 3 false positives in sub-images 5 and 13. This means that in these sub-images, the model predicted 3 objects as trichomes, but they were not actually trichomes. Additionally, there were 53 false negatives, indicating that the model failed to predict 53 objects that were indeed trichomes. The tuning parameters used in this study were 100 epochs, a learning rate of 0.01, a batch size of 16, and an image size of 640. Tuning parameters is crucial to enable trial and error until the best model is found. After 100 iterations (epochs) of training, the mean average precision (mAP) value at a confidence threshold of 50 was 0.816. The mAP value at confidence thresholds of 50 - 90 was 0.38. Details of the loss metrics are as follows: box loss was 1.408, classification (cls) loss was 0.8149, and distribution focal loss (dfl) is 0.9129.

This study has shown that the deep learning model employed has significant potential for detecting trichomes, particularly glandular bulbous trichomes. However, to enhance detection accuracy and efficiency, several future development directions need to be explored. First, researchers could consider utilizing more advanced deep learning models, such as deeper CNNs or the latest architectures like EfficientNet, which have proven effective in various image recognition tasks. These models could provide improved performance in terms of accuracy and processing speed. Second, implementing hybrid methods that combine deep learning techniques with traditional image processing algorithms may be a solution to enhance detection outcomes. By leveraging the strengths of both approaches, the model can become more adaptable to variations in data, such as changes in lighting conditions and image angles. Additionally, the findings of this research have broad practical implications, particularly in the field of precision agriculture. With the ability to accurately detect trichomes, this technology can assist farmers in monitoring plant health and predicting yield outcomes. This not only enhances efficiency in crop management but also contributes to reducing unnecessary pesticide and fertilizer use, thereby supporting more sustainable agricultural practices.

References

- [1] Y. Zhang *et al.*, "The Roles of Different Types of Trichomes in Tomato," *Agronomy*, Vol. 10, No. 3, , 2020, Art. n. 411, doi: 10.3390/agronomy10030411.

- [2] N. Bergau, S. Bennewitz, F. Syrowatka, G. Hause, and A. Tissier, “The Development of Type VI Glandular Trichomes in the Cultivated Tomato *Solanum Lycopersicum* and a Related Wild Species *S. Habrochaites*,” *BMC Plant Biology*, Vol. 15, No. 1, pp. 1–15, 2015, doi: 10.1186/s12870-015-0678-z.
- [3] J.J. Glas, B.C.J. Schimmel, J.M. Alba, R. Escobar-Bravo, R.C. Schuurink, and M.R. Kant, “Plant glandular trichomes as targets for breeding or engineering of resistance to herbivores,” *International Journal of Molecular Sciences* Vol. 13, No. 12, pp. 17077–17103, 2012, doi: 10.3390/ijms131217077.
- [4] A.F. Lucatti, A.W. van Heusden, R.C.H. de Vos, R.G.F. Visser, and B. Vosman, “Differences in Insect Resistance Between Tomato Species Endemic to the Galapagos Islands,” *BMC Evolutionary Biology*, Vol. 13, No. 1, 2013, Art. No. 175, doi: 10.1186/1471-2148-13-175.
- [5] X. Ni, C. Li, H. Jiang, and F. Takeda, “Deep Learning Image Segmentation and Extraction of Blueberry Fruit Traits Associated with Harvestability and Yield,” *Horticulture Research*, Vol. 7, 2020, Art. No. 110, doi: 10.1038/s41438-020-0323-3.
- [6] S.M. Narkhede *et al.*, “Machine Learning Identifies Digital Phenotyping Measures Most Relevant to Negative Symptoms in Psychotic Disorders: Implications for Clinical Trials,” *Schizophrenia Bulletin*, Vol. 48, No. 2, pp. 425–436, 2022, doi: 10.1093/schbul/sbab134.
- [7] B. Vosman *et al.*, “QTL Mapping of Insect Resistance Components of *Solanum Galapagense*,” *Theoretical and Applied Genetics*, Vol. 132, No. 2, pp. 531–541, 2019, doi: 10.1007/s00122-018-3239-7.
- [8] T. Diwan, G. Anirudh, and J.V. Tembhurne, “Object Detection Using YOLO: Challenges, Architectural Successors, Datasets and Applications,” *Multimedia Tools and Applications*, Vol. 82, No. 6, pp. 9243–9275, 2023, doi: 10.1007/s11042-022-13644-y.
- [9] I. Denata, T. Rismawan, and I. Ruslianto, “Implementation of Deep Learning for Classification Type of Orange Using The Method Convolutional Neural Network,” *Telematika*, Vol. 18, No. 3, pp. 297–307, 2021, doi: 10.31315/telematika.v18i3.5541.
- [10] S. Indolia, A.K. Goswami, S.P. Mishra, and P. Asopa, “Conceptual Understanding of Convolutional Neural Network-A Deep Learning Approach,” *Procedia Computer Science*, Vol. 132, pp. 679–688, 2018, doi: 10.1016/j.procs.2018.05.069.
- [11] J.P. Onnela, “Opportunities and Challenges in the Collection and Analysis of Digital Phenotyping Data,” *Neuropsychopharmacology*, Vol. 46, No. 1, pp. 45–54, 2021, doi: 10.1038/s41386-020-0771-3.
- [12] M.H. Saleem, J. Potgieter, and K.M. Arif, “Plant Disease Classification: A Comparative Evaluation of Convolutional Neural Networks and Deep Learning Optimizers,” *Plants*, Vol. 9, No. 10, pp. 1–17, 2020, doi: 10.3390/plants9101319.
- [13] R. Kaur and S. Singh, “A Comprehensive Review of Object Detection With Deep Learning,” *Digital Signal Processing*, Vol. 132, 2022, doi: 10.1016/j.dsp.2022.103812.
- [14] J. Liu and X. Wang, “Plant Diseases and Pests Detection Based on Deep Learning: A Review,” *Plant Methods*, Vol. 17, No. 1, pp. 1–18, 2021, doi: 10.1186/s13007-021-00722-9.
- [15] W. Albattah, M. Nawaz, A. Javed, M. Masood, and S. Albahli, “A Novel Deep Learning Method for Detection and Classification of Plant Diseases,” *Complex & Intelligent Systems*, Vol. 8, no. 1, pp. 507–524, 2022, doi: 10.1007/s40747-021-00536-1.
- [16] S.S. Harakannanavar, J.M. Rudagi, V.I. Puranikmath, A. Siddiqua, and R. Pramodhini, “Plant Leaf Disease Detection Using Computer Vision and Machine Learning Algorithms,” *Global Transitions Proceedings*, Vol. 3, No. 1, pp. 305–310, 2022, doi: 10.1016/j.gltp.2022.03.016.
- [17] L. Li, S. Zhang, and B. Wang, “Plant Disease Detection and Classification by Deep Learning - A Review,” *IEEE Access*, Vol. 9, pp. 56683–56698, 2021, doi: 10.1109/ACCESS.2021.3069646.
- [18] C. Chen *et al.*, “YOLO-Based UAV Technology: A Review of the Research and Its Applications,” *Drones*, Vol. 7, No. 3, 2023, doi: 10.3390/drones7030190.
- [19] J. Huixian, “The Analysis of Plants Image Recognition Based on Deep Learning and Artificial Neural Network,” *IEEE Access*, Vol. 8, pp. 68828–68841, 2020, doi: 10.1109/ACCESS.2020.2986946.
- [20] L. Kang, Z. Lu, L. Meng, and Z. Gao, “YOLO-FA: Type-1 fuzzy attention based YOLO detector for vehicle detection,” *Expert Systems with Applications*, Vol. 237, 2024, doi: 10.1016/j.eswa.2023.121209.
- [21] J.W. Wu, W. Cai, S.M. Yu, Z.L. Xu, and X. Y. He, “Optimized Visual Recognition Algorithm in Service Robots,” *International Journal of Advanced Robotic Systems*, Vol. 17, No. 3, pp. 1–11, 2020, doi: 10.1177/1729881420925308.