NHL STENDEN
computer vision & data science

Two-Stage Lettuce Detection and Classification

NHL Stenden Lectoraat in Computer Vision & Data Science

Veronika Jadrná

Supervisor: Willem Dijkstra

Abstract— Monitoring crop development process is one of the key aspects in farming, as environmental conditions greatly influence plant growth and overall productivity, especially for sensitive crops like lettuce. This study offers an innovative approach to monitor lettuce plants and provide farmers with real-time insights into growing patters, which are imperative for harvest prediction and resource allocation. The methodology consists of a two-stage approach for detection and classification of lettuce plants into three groups - small, medium, large. The study employed YOLOv5s model for a single class object detection and tested and evaluated three different classification methods. The YOLOv5s model achieved impressive results in detecting individual lettuce plants varying in sizes. The best-performing classification method was a distance-based classification, which calculates the distance of a diagonal value of the bounding box to the statistical metrics of each growth stage. The results of this system highlight it´s potential for real-world farming practices, offering a significant improvements while saving time and resources.

1 Introduction

While often overlooked and undervalued, the contributions of the farming community are crucial to the functioning of our global food systems. Yet, the resources available to farmers are often insufficient to meet the demands they face. With growing challenges such as climate change, labor shortages and increased demand from a rising population it is clear, that optimization of agricultural practices is no longer a choice, but a necessity [1].

One of the key aspects in farming is understanding the crop development process. Environmental factors such as temperature and precipitation play a significant role in influencing the growth cycle. Since the plant growth has a strong impact on agricultural productivity, this can lead to severe financial consequences for farmers [2]. Monitoring stages of plant growth can bring many benefits for farmers and help them make informed decisions on time. This is especially important for crops like lettuce, which are very sensitive to environmental fluctuations [3].

Lettuce, known for its low-calorie content and rich nutrition including fibre, vitamins and minerals, also offers health benefits such as the reduction of cardiovascular disease and other chronic conditions [4]. In addition to these health benefits, lettuce holds an economic importance in the agricultural sector as it is one of the most consumed vegetables worldwide.

Current methods of monitoring lettuce are time-consuming and inefficient. Approaches such as manual inspection and analysis of lettuce plants lack the speed and precision necessary for optimizing lettuce production. This is where modern technology can make a significant impact.

As highlighted in previous research [5], the use of computer vision in agriculture addresses these issues by automating the process, providing real-time data and reducing human error. Furthermore, this technology allows for more efficient monitoring of crop development and enables farmers to detect issues on time. With the help of these technologies, current agricultural practices can experience significant improvements in both efficiency and productivity, ultimately leading to higher yields and better crop quality.

- *Veronika Jadrná is a Computing Science student at the NHL Stenden University of Applied Sciences, E-mail: veronica.jadrna@student.nhlstenden.com.*
- *Willem Dijkstra is a lector at the NHL Stenden Lectoraat in Computer Vision & Data Science, E-mail: willem.dijkstra@nhlstendend.com.*

1.1 State of the Art

Recently, the use of computer vision and machine learning in agriculture has experienced rapid growth. There have been many studies conducted for optimizing farming practices.

One of the studies in this area explored the use of various machine learning algorithms such as Logistic Regression (LR), K-Nearest Neighbors (KNN) and Linear Support Vector Machines (L-SMV) to classify growth stages of lettuce grown in a smart aquaponics system [6]. In the study, the machine vision techniques were used to process images of lettuce plants using methods such as superpixels, k-means clustering and watershed segmentation for feature extraction. By extracting the features, they aimed to accurately classify different stages of lettuce and divided them into three groups: vegetative, head development and harvest. Subsequently three machine learning models were trained using the processed dataset. Each model was optimized using GridSearchCV to determine the best parameters. The KNN model achieved the best performance compared to the other models with a classification accuracy of 91.67%.

Another study from 2020 [7] further explored the use of CNNs for image classification and analysis across various crop growth stages. One aspect of their research included monitoring lettuce growth. The CNN models were used to identify and separate plants from the background for a more accurate measurement of the canopy area, which is a key factor often used to analyze and track overall plant growth. By estimating growth indicators such as leaf weight, this study contributed to improved tracking of crop development.

Furthermore, in a more recent study [8] a method for identifying the growth status of hydroponic lettuce was developed. The researchers used a YOLO-EfficientNet system which combined the YOLO-v8n model for lettuce detection and the EfficientNet-v2s model for classifying the growth stages into three categories: healthy, diseased and pest-infested. The model demonstrated impressive performance with a 99% accuracy in detecting lettuce plants using YOLO-v8n and 94.68% test accuracy with the EfficientNet-v2s. The impressive F1-score of 96.18% further highlighted the system´s effectiveness and accuracy.

An earlier study [9], focused on developing computer vision system to monitor tomato growth stages. In order to detect tomato flowers and fruits, two object detection models were implemented, the Regional-based Convolutional Neural Network (R-CNN) and Single Shot Detector (SSD). The SSD vastly outperformed the R-CNN with an accuracy of 100% for flower detection and 95.99% for fruit detection. Furthermore, this study also involved the implementation of machine learning models such as Artificial Neural Network (ANN), K-Nearest Neighbours (KNN), and Support Vector

Machine (SVM) for maturity grading of tomato fruits. The most effective was the SVM model with a training-testing accuracy rate ranging from 97.78% to 99.81%.

1.2 Research Objectives

In this research, we focus on applying computer vision and machine learning models to improve and automate lettuce crop monitoring. The primary objective is to detect lettuce at various growth stages and assess their individual sizes by measuring plant diameter. The YOLO (You Only Look Once) object detection algorithm will be utilized in this research due to its efficiency in detecting multiple objects within a single frame, making it suitable for monitoring lettuce plants throughout their growth cycles.

By measuring the bounding boxes generated by YOLO, the growth stages of the crops can be classified based on their size. The use of bounding boxes enables more precise measurement and helps identify underperforming or potentially unhealthy crops.

This system provides farmers with real-time insights into the development of individual plants and assists them with harvest prediction. Additionally, this approach enables for better resource management such as optimizing water supply based on the current growth phase.

This research aims to address the following questions:

- How accurately can the YOLO model detect lettuce plants at different stages of growth?
- What is the best method for classifying the lettuce plants based on the diagonal sizes of their bounding boxes?
- Is the model able to handle challenge of overlapping leaves, particularly in the later growth stages?

By answering these questions, the study wants to contribute to the development of more efficient and reliable solutions in crop monitoring techniques and benefit the agricultural industry.

2 Materials and Methods

This section delves into data preparation process and methodology used within this research.

2.1 Video Dataset

The dataset used in this research was obtained in an open agricultural field. As shown in Figure 1, a camera was attached to a farming vehicle, which moved along the rows off lettuce plants and captured continuous video footage of the field. The camera was positioned to capture clear images of the lettuce plants from above and at a close distance to ensure plants of all sizes were visible. The videos captured lettuce plants at various growth stages, including early-stage seedlings, mid-stage, and fully grown lettuce ready for harvest.



Fig. 1. A farming vehicle capturing video footage of lettuce in the field.

In total, eight videos were recorded with duration ranging from 2 to 14 minutes. The frame rate was maintained at approximately 30 frames per second for all videos. The dataset includes videos with a resolution of 1920 x 1080, a standard Full HD format. While the file properties report the mentioned resolution, this corresponds to the default metadata format. Most of the videos are viewed in portrait mode and are 1080 pixels wide and 1920 pixels high.

2.2 Preprocessing and Frame Extraction

To prepare the video data for model training, individual frames were extracted from the videos. The purpose of extracting frames was to convert video footage into a collection of still images ready to be used for object detection tasks. The extraction process was carried out for each video and the images were saved in corresponding output folder. As each video predominantly captured lettuce at different growth stage, organizing the frames into separate folders allowed for clear separation and distinction between the growth stages of lettuce.

The frames were extracted at a set interval of 1 frame per second, resulting in a total of 3,338 frames. By setting this specific rate, the redundancy was minimized. This effectively reduced the overlap between frames and the likelihood of including nearly identical pictures multiple times.

The extracted frames had resolutions based on the orientation of the original videos. The video dataset consisted of both, portrait and landscape videos. The majority of the videos had resolution of 1080 x 1920 pixels. The extracted frames from the landscape videos were rotated during preprocessing to ensure all images are uniform before starting the annotation process. Figure 2 shows example frames from the dataset, illustrating the variety of lettuce growth stages captured during extraction.



Fig. 2. Example frames depicting different growth stages.

2.3 Data Annotation

The preprocessed images were annotated using a single class “lettuce” and bounding boxes drawn around each lettuce plant as seen in Figure 3 .

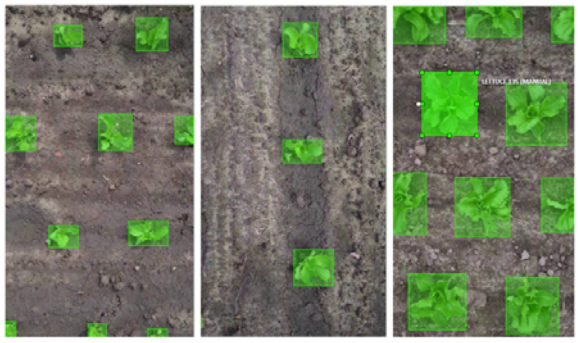


Fig. 3. Example annotated frames with a single class ”Lettuce”.

In total 140 images were manually annotated. However, fully grown lettuce plants, such as the one in the provided Figure 2, were not labeled due to the dense, overlapping foliage that would lead to imprecise and inaccurate annotations. Despite this, this research aims to test the images of fully grown lettuce in the experimental part of the research to see how the model handles these types of images.

2.4 YOLOv5

YOLOv5 is an advanced version of YOLO (You Only Look Once) series for real-time object detection developed by Ultralytics. In comparison to previous models this model is optimized for speed and accuracy and uses advanced neural network architecture built with PyTorch [10]. One of its key enhancements is the use of Cross Stage Partial Networks (CSPNet) for feature extraction and Spatial Pyramind Pooling (SPP) for further enhancing feature extraction and maintaining efficiency. Moreover, the use of Path Aggregation Networks (PANet) improves detection especially for objects at different scales [11] [12].

The YOLOv5 comes in five versions: n, s, m, l and x, each offering a balance between complexity and detection performance. These models can be used for different purposes and demands, from low-power devices to high-performance environments. Table 1 shows an overview of different model variants, inference speed on GPU and CPU and parameter counts for a 640-pixel image.

Table 1. YOLOv5 Model Version Overview [12]

Model Version	Parameters (Million)	mAP 0.5 (%)	CPU Time (ms)	GPU Time (ms)
YOLOv5n	1.9	45.7	45	6.3
YOLOv5s	7.2	56.8	98	6.4
YOLOv5m	21.2	64.1	224	8.2
YOLOv5l	46.5	67.3	430	10.1
YOLOv5x	86.7	68.9	766	12.1

2.5 Architecture

This work utilizes implementation of YOLOv5 from Ultralytics [10] depicted in Figure 4. The architecture consists of three main parts: the backbone, neck and head.

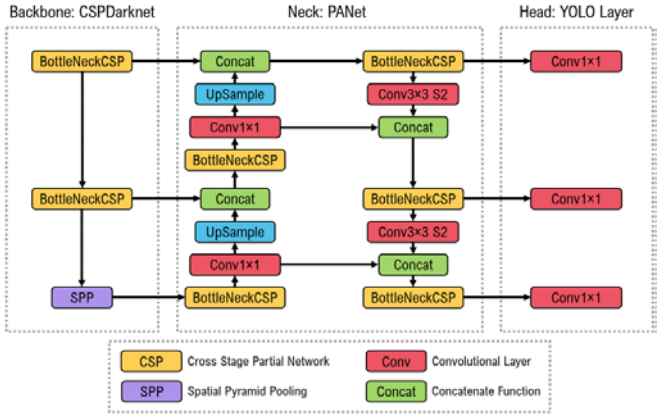


Fig. 4. YOLOv5 Architecture [13].

The backbone utilizes CSPDarknet53 which incorporates Cross Stage Partial Network (CSPNet). It´s design includes Conv+BatchNorm+SiLU (CBS) modules and C3 modules, which improve feature extraction from input images by enabling better gradient flow and minimizing duplicate information [11]. This combination of components helps the network maintain high speed and accuracy when processing complex images.

In the neck component of YOLOv5 architecture we can see a variant of Spacial Pyramid Pooling (SPP) and the Path Aggregation Network (PANet) modified by applying BottleNeckCSP in it´s architecture. The PAN is enhancing the flow of information and helps to locate pixels in the task of mask prediction. The SPP is responsible for aggregation of information received from the inputs and returning of a fixed length output. This structure is designed to make the model accurately detect objects of different sizes and improve the speed of the network. [14]

The final head of YOLOv5 applies the detection process. It outputs the bounding boxes and class predictions for each object in the image. Predefined bounding boxes are adjusted during training to match the dimensions of object in the dataset thanks to the use of anchor-based method. Anchor boxes act as fixed reference boxes that the network adjusts for the correct size and location based on the detected object. Furthermore, the model employs a loss function that combines classification and localization loss to further improve accuracy [15] [12].

2.6 Yolov5 Training

The YOLOv5s model was employed in this project to perform single class object detection. The dataset was split as follows: 70% training, 15% validation and 15% testing. In total 140 images were used out of which 98 were used for training.

During training, a tiled approach is used to effectively manage memory usage and improve detection of smaller objects. The tile size of 512x512 is used during training and validation with a batch size of 3 and 4 tiles per image. During testing the images are displayed in their original dimensions.

Data preprocessing for training included augmentations such as horizontal flips and rotations to further improve the performance. For loading the data during training and validation, a custom data loader is used that supports random positive tiling. This technique ensures that each tile contains at least one positive sample and avoids using tiles with no object. The data is shuffled before each epoch to prevent learning bias from fixed data ordering.

Optimization was performed using Adam Optimizer with a learning rate 0.0001. A learning rate scheduler is used to dynamically reduce the learning rate in case the validation loss starts to stagnate. Validation is conducted every 2 epochs and the model with the best validation loss is saved. Training metrics including loss and validation performance are visualized in TensorBoard.

2.7 Dynamic Range Classification Using Mean and Standard Deviation

The first method used to classify the growth stage of the lettuce in this project was based on dynamically calculated ranges. The mean and standard deviation of the diagonal lengths of the bounding boxes were the main statistical metrics to set the ranges. The diagonal length of the bounding boxes was calculated using the formula:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

where (x_1, y_1) and (x_2, y_2) are the coordinates of the top-left and bottom-right corners of the bounding box. For each growth stage, the mean and standard deviation were calculated using the diagonal lengths and used to define classification ranges as follows:

$$\text{Lower Threshold} = \mu_{\text{stage}} - k\sigma_{\text{stage}}, \quad (1)$$

$$\text{Upper Threshold} = \mu_{\text{stage}} + k\sigma_{\text{stage}}. \quad (2)$$

where μ_{stage} and σ_{stage} represent the mean and standard deviation of diagonal lengths for a specific growth stage, and k is a tunable multiplier that adjusts the standard deviation and therefore the width of the classification range. A smaller value of k results in narrower ranges for each growth stage, and conversely, a larger value of k results in broader ranges.

After the diagonal of the bounding box is calculated, a lettuce plant is classified as small, medium or large, based on the range it falls into. For instance, a bounding box diagonal that lies within the range $[\mu_{\text{small}} - k\sigma_{\text{small}}, \mu_{\text{small}} + k\sigma_{\text{small}}]$ is classified as small, while diagonals that fall within the corresponding ranges for medium or large are classified into those stages.

2.8 Fixed Threshold Classification

Another approach that will be explored during this project is classifying lettuce into three groups – small, medium, big – based on the fixed threshold values. This method uses static, predefined boundaries to separate growth stages. After the diagonal lengths are calculated the lettuce is classified based on the range it falls into. The classification is performed as follows:

$$\begin{aligned} \text{Small}: &\quad d < T_1, \\ \text{Medium}: &\quad T_1 \leq d < T_2, \\ \text{Big}: &\quad d \geq T_2. \end{aligned}$$

Here the T1 and T2 represent fixed thresholds to separate the growth stages. The bounding boxes with diagonals that are smaller than T1 will be classified as small while those between T1 and T2 are classified as medium. Lastly the diagonals that are larger than T2 are classified as big. This method provides a simpler approach in classifying the lettuce plants based on the size of their diagonals.

2.9 Distance-Based Classification

The third approach for classifying bounding boxes into three different growth stages is based on calculating the distance of each diagonal to the mean and standard deviation values of each stage. The predefined mean and standard deviation values are obtained from prior analysis of the dataset and represent the typical diagonal length and variability within each growth stage. The normalized distance is calculated using the following formula:

$$\text{Distance}_{\text{stage}} = \frac{|d - \mu_{\text{stage}}|}{\sigma_{\text{stage}}}$$

Here, d represents the diagonal length, μ_{stage} is the mean of a specific stage, and σ_{stage} is the standard deviation of a specific stage.

After the distance is calculated for each stage, the bounding box is classified as the growth stage with the smallest normalized distance. This approach ensures that the bounding box is classified into the most likely growth stage by taking into account the statistical metrics.

3 Experiments and Results

This section provides an overview of the experiments and results that were conducted for this research in order to address the research questions. The primary focus of the experiments was to find the best method for classifying the growth status of the lettuce plants and test the performance of the YOLOv5 pipeline on fully grown lettuce plants with a significant overlap.

3.1 Object Detection Performance

In this study YOLOv5s model was employed for a single class object detection task. The model demonstrated impressive object detection performance across the dataset, effectively identified bounding boxes and achieved high metrics.

The tuning of the hyperparameters played an important role in optimizing the model´s performance. The most influential hyperparameter during the research was the learning rate. The model struggled to predict bounding boxes with learning rate 0.001, however, lowering the learning rate to 0.0001 resulted in significant improvements.

To further improve detection performance, additional experiments were conducted including applying augmentation technique. The comparison of model´s performance with and without the augmentation applied is shown in Table 2.

Table 2. Impact of Data Augmentation on Detection Performance

Augmentation	F1 Score	Precision	Recall	mAP
No	0.96	0.93	0.99	0.89
Yes	0.98	0.98	0.99	0.90

Another comparison, that can be seen in the Table 3, was done to define the best tile sizes with the data augmentation applied. In order to preprocess the dataset containing both, small and large objects, in this case lettuce plants, medium tile sizes – 512x512 and 640x640 – were compared and evaluated.

Table 3. Tile Size Comparison for Detection Performance

Tile Size	F1 Score	Precision	Recall	mAP
640x640	0.97	0.96	0.99	0.88
512x512	0.98	0.98	0.99	0.90

After evaluating the results the research decided to implement augmentation, and continue with tile size 512x512 as it achieved better scores than the 640x640 tile size. The model achieved an overall mean Average precision of 90% and precision of 98% at LoU threshold of 0.5.

Example output images of the YOLOv5 detection model can be seen in Figure 5.



Fig. 5. Example output images with YOLOv5-generated white bounding boxes.

From the output images we can see that the model successfully generated white bounding boxes with a class ”Lettuce” around each individual and differently sized plant.

3.2 Classification

After YOLOv5 was implemented and fine-tuned for single class object detection, the classification logic was implemented in this pipeline. All of the classification methods were based on the diagonal lengths of the bounding boxes that were used to classify lettuce as small, medium or large depending on their sizes. The experiments were done in a separate script to perform the analysis before applying and testing a specific method with the model. The research tested and compared three main classification methods:

- Dynamic Range Classification
- Fixed Pixel Threshold Classification
- Distance-Based Classification

3.2.1 Dynamic Range Classification

This method used a statistical approach by using the mean and standard deviation of each growth stage to dynamically define the classification ranges. The mean and standard deviation as well as the minimum and maximum diagonal value of each growth stage can be seen in Table 4.

Table 4. Summary statistics for different categories.

Category	Min	Max	Mean	Std
Small	90.69	224.51	148.57	27.25
Medium	87.66	361.01	250.24	67.07
Big	233.00	514.24	386.03	73.15

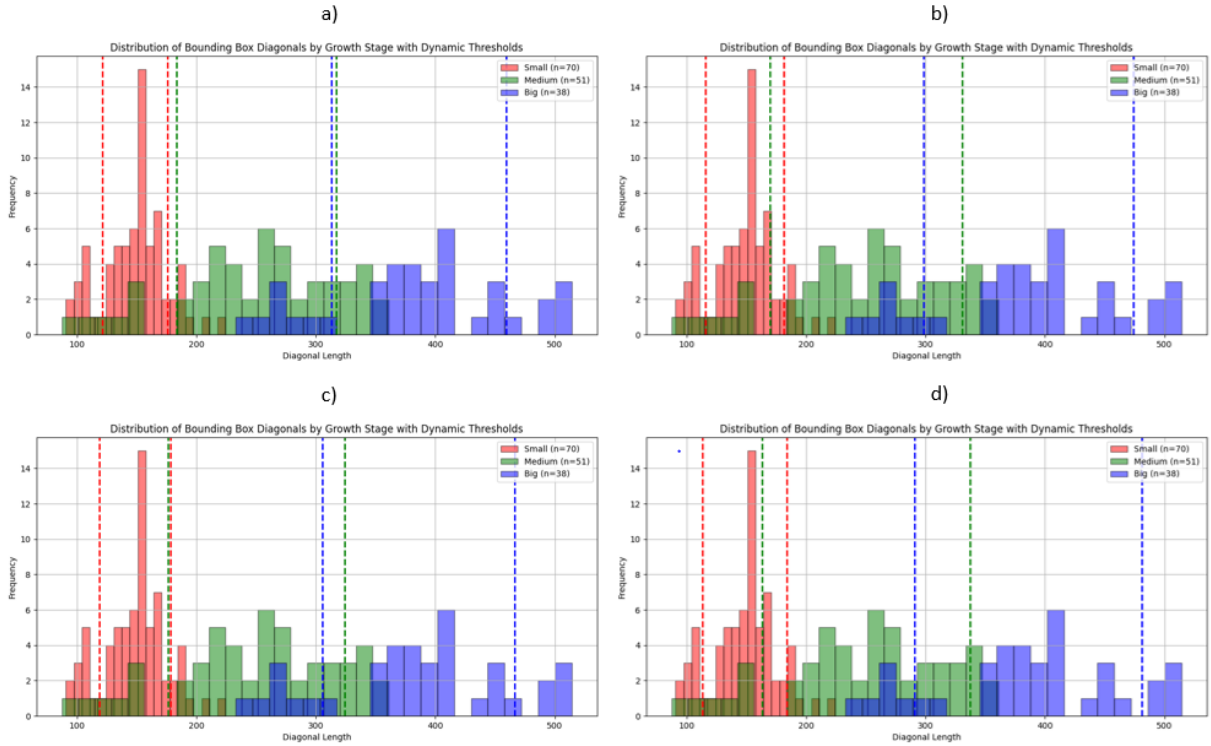


Fig. 6. Distribution of diagonals for each growth stage with predicted ranges using different values of k. a) $k = 1$, b) $k = 1.2$, c) $k = 1.1$, d) $k = 1.3$.

Using these statistical metrics, the lower threshold 1 and upper threshold 2, which can be seen in section 2.7, are defined to set the ranges for each growth stage. The distribution of the diagonals for each stage as well as the predicted ranges can be seen in Figure 6 . Here we can see how the ranges are defined using a dashed line and a specific color for each growth stage.

Even after testing different values of k to widen or narrow the ranges, the overlaps and gaps are still present. Gaps represent the data that would not be classified at all and the overlaps highlight the area where the bounding box would fall within two classes. That is why this approach is not suitable for our dataset.

3.2.2 Fixed Threshold Classification

Another method tested within this research, employs a simpler approach in classifying the lettuce plants by setting a fixed pixel thresholds and creating specific ranges for different growth stages.

The performance of this method was evaluated by testing different threshold combinations for separating bounding box diagonals into small, medium and large growth stages. In Table 5 we can see examples of randomly selected combinations of the threshold values and their results.

After analyzing the metrics, the best performing threshold combination with the least misclassifications was:

- Small: Diagonal length $d < 192$,
- Medium: Diagonal length $192 \leq d < 344$,
- Large: Diagonal length $d \geq 344$.

A careful selection of threshold values is crucial for minimizing the number of misclassifications. While this method ensures consistency, the reliance on predefined thresholds makes it less adaptable to datasets with significant variability or overlap between growth stages.

Small	Medium	Large	Total Misclassified
<210	210 - 345	>345	25
<230	230 - 360	>360	32
<220	220 - 350	>350	27
<215	215 - 345	>345	25
<210	210 - 340	>340	26
<215	215 - 340	>340	26
<212	212 - 345	>345	24
<212	212 - 350	>350	24
<212	212 - 355	>355	24
<212	212 - 352	>352	25
<212	212 - 351	>351	24
<208	208 - 351	>351	24
<196	196 - 351	>351	24
<192	192 - 351	>351	24
<192	192 - 348	>348	24
<192	192 - 344	>344	23

Table 5. Summary of classification ranges and total misclassified counts.

3.2.3 Distance-Based Classification

This approach implements the calculation of a normalized distance. Growth stages are assigned based on the smallest normalized distance between the bounding box diagonal and stage-specific statistical values.

In order to improve accuracy of these statistics, a margin exclusion technique was implemented. This technique aimed to solve a problem with the bounding boxes located at the edge of the image. These bounding boxes introduced smaller diagonal lengths that influenced the statistical parameters (mean and standard deviation) for each growth stage. In order to achieve even more accurate classification these partially visible plants were excluded using a margin of 5 pixels. In Figure 7, we can see a comparison of diagonal distribution without and with the margin exclusion.

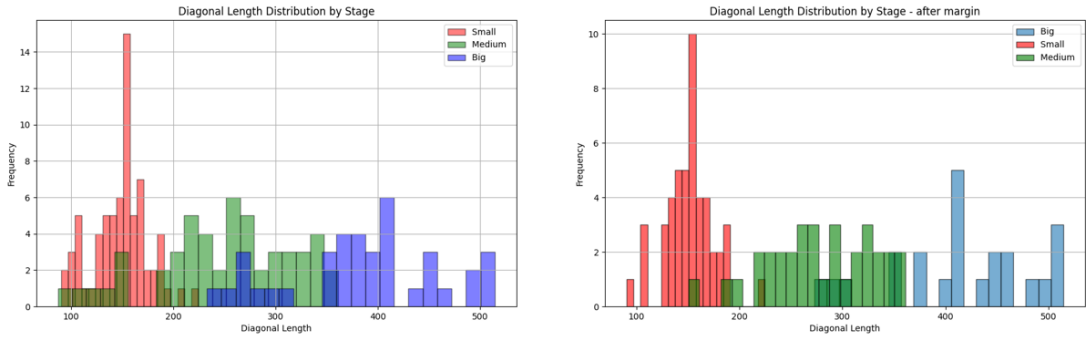


Fig. 7. Diagonal length distribution before and after applying the margin technique.

This margin technique ensured the diagonals were calculated only for those bounding boxes that were capturing fully visible plants.

After excluding the bounding boxes that were on the edge of the image, the overlap was significantly reduced, leading to more accurate mean and std values. Table 6 shows a comparison of the mean and standard deviation before, and after applying the margin exclusion technique.

Table 6. Statistics before and after margin exclusion technique.

	Before Margin Exclusion		After Margin Exclusion	
Stage	**Mean**	**Std Dev**	**Mean**	**Std Dev**
Small	148.57	27.25	153.37	26.55
Medium	250.24	67.07	288.65	52.22
Large	386.03	73.15	422.08	62.99

The results after applying margin exclusion technique were significantly improved as seen in Table 7. The classification achieved precision of 94% for small lettuce, 88% for medium lettuce, and 91% for large lettuce.

Table 7. Precision and F1 score before and after applying margin exclusion technique.

	Before Margin Exclusion		After Margin Exclusion	
Stage	**Precision**	**F1 score**	**Precision**	**F1 score**
Small	0.90	0.88	0.94	0.96
Medium	0.67	0.67	0.88	0.86
Large	0.75	0.77	0.91	0.89

In Figure 8, we can see example output images from the YOLOv5 model with the distance-based classification applied. The red color of the bounding box is used for small lettuce, green for medium and blue for large lettuce plants.

As seen from the pictures, some lettuce plants were assigned a different class due to their size difference compared to the rest of the plants in the particular stage. More detailed visual comparison of true and false classifications can be seen in Figure 9 .

Misclassifications occurred when some plants differed significantly in size from other plants within their group, resulting in being categorized into a different stage. However, in most cases, such classifications may not truly be errors. For example in Figure 9, small lettuce is classified as medium, but, it´s size truly is larger than the rest of the small lettuce plants. Similarly, medium lettuce is classified as small and large lettuce as medium. However, their sizes are much smaller compared to the rest within their growth stage, resulting in being classified as one stage below.

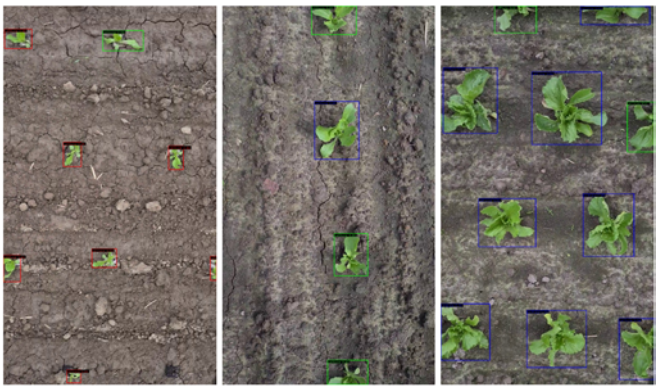


Fig. 8. Example output images of lettuce plants after the classification, where red bounding box is used for small, green for medium and blue for large lettuce.

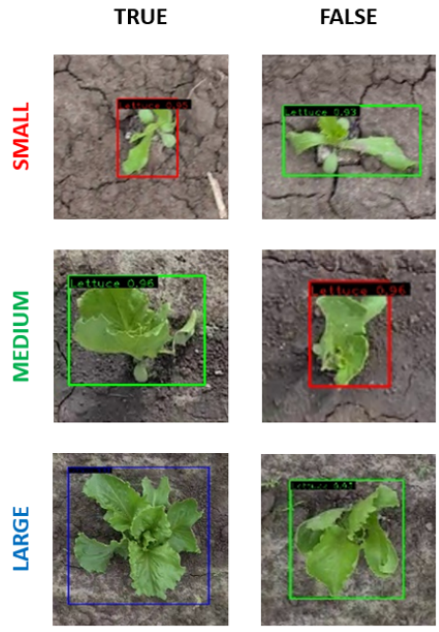


Fig. 9. Visual comparison of true and false classification examples for each growth stage, where red color of bounding box is used for small lettuce, green for medium and blue for large.

3.3 Detection of Fully Grown Lettuce

Lastly an experiment with fully grown lettuce plants was conducted. As described in the materials and methods part of this paper, the images of fully grown lettuce were excluded from the dataset used for this project due to their dense and overlapping content. These images were not labeled as it would lead to incorrect annotations. However, the research aimed to test these images as a part of experiments to see how the model performs and handles this type of images.

In Figure 10, we can see the output images of fully grown lettuce.



Fig. 10. Example output images of fully grown lettuce plants.

As expected, the model was confused by these images since they were not part of the training dataset and failed to generate correct predictions.

4 Discussion

This study addressed and utilized a two-stage approach for detection and classification of lettuce plants into three main growth stages – small, medium, large. The methodology used, ensured a seamless transition from detection of different lettuce plants and their classification based on diagonal values of their bounding boxes.

The YOLOv5 model employed in this research, achieved a high accuracy in detecting individual lettuce plants varying in sizes, and successfully generated bounding boxes that served as a base for classification process. This model achieved a recall of 99% and precision of 98%, proving its ability to accurately identify lettuce plants.

For the classification of lettuce plants, three methods were evaluated. As seen from the results, the dynamic range classification that used standard deviation and mean of each growth stage to find the boundaries did not perform well with our dataset. Even after experimenting with different values of k, the multiplier used to narrow or broaden the ranges for each stage, this method struggled with overlapping ranges and gaps that would lead to inconsistent results. The overlaps represent the lettuce that would fall within two stages and the gaps would cause some lettuce to not be classified at all.

Another method that has been used in this study to classify lettuce plants into three growth stages was simple fixed thresholding. While this method performed better than the dynamic range classification, its reliance on static thresholds makes it unable to adjust to different grown patterns and overlaps in the dataset.

Lastly, distance-based method for classifying the lettuce plants was implemented. Unlike the previous methods, the distance-based classification accounts for variability in the dataset. This method continuously evaluates the likelihood of each stage relative to the statistical metrics of the data. The comparison of distances across all stages ensures, that the most appropriate stage is used to classify the lettuce plant even in ambiguous scenarios. Therefore, this method handles the overlaps between data more accurately. Furthermore, the chance of a plant being misclassified was significantly reduced after the margin exclusion technique was applied. The classification achieved precision of 94% for small lettuce, 88% for medium lettuce, and 91% for large lettuce.

While the YOLOv5 model demonstrated strong performance in detecting lettuce plants, and the distance-based method in classifying them, further improvements should be made. One challenge arises when plants overlap significantly. Since this study did not include the overlapping lettuce plant images in the training data the model, as expected, failed to detect them. For the future work the dataset could be refined to include such examples in the dataset and explore advanced technique for detection of densely overlapping plants. While the research applied margin exclusion technique to not include these in the statistical metrics for distance-based classification, they were not removed from output images. Removing or filtering these cases out could improve the overall visualization and ensure only fully visible plants are prioritized.

5 Conclusion

This study highlights the importance of monitoring plant growth status as a crucial factor in modern farming and offers an innovative approach to assist farmers in managing agricultural productivity. By introducing a YOLOv5-based system for detection and classification of lettuce plants into three main stages – small, medium, large – this study addresses and automates key agricultural practices. The integration of object detection and classification within a single model simplifies the process and makes it suitable for real-world application.

The model has proved its efficacy in detecting the lettuce plants with a recall of 99% and F1 score and precision of 98%. The results demonstrate the model´s ability to monitor crops under different conditions including varied plant sizes and complex backgrounds. Furthermore, the distance-based method successfully classified small lettuce with a precision of 94% , medium lettuce with a precision of 88%, and large lettuce with a precision of 91%.

This system offers farmers real-time insights into growing patterns which are imperative for harvest planning and resource allocation such as adjusting water supply or crop fertilization. Furthermore, the growing patterns might discover possible issues among the plants that are calling for action. By introducing this model, the study aimed to create a tool that could be integrated in real-world farming practices and lead to improvements, while saving time and resources.

6 Acknowledgments

This project is financially supported by the European Union within the JTF-call project DigiAgro3.

References

[1] Rajveer Dhillon and Qianna Moncur. Small-scale farming: A review of challenges and potential opportunities offered by technological advancements. *Sustainability*, 15(21):15478, 2023.

[2] Vasyl Cherlinka. Stages of plant growth and their role in agriculture. *EOS Data Analytics Blog*, 2024. Last updated: 11.04.2024.

[3] Alan Bauer, Aaron George Bostrom, Joshua Ball, Christopher Applegate, Tao Cheng, Stephen Laycock, Sergio Moreno Rojas, Jacob Kirwan, and Ji Zhou. Combining computer vision and deep learning to enable ultra-scale aerial phenotyping and precision agriculture: A case study of lettuce production. *Horticulture Research*, 6:70, 2019.

[4] Moo Jung Kim, Youyoun Moon, Janet C. Tou, Beiquan Mou, and Nicole L. Waterland. Nutritional value, bioactive compounds and health

benefits of lettuce (lactuca sativa l.). *Journal of Food Composition and Analysis*, 49:19–34, 2016.

[5] Hongkun Tian, Tianhai Wang, Yadong Liu, Xi Qiao, and Yanzhou Li. Computer vision technology in agricultural automation—a review. *Information Processing in Agriculture*, 7(1):1–19, 2020.

[6] Sandy C. Lauguico, Ronnie S. II Concepcion, Jonnel D. Alejandrino, Rogelio Ruzcko Tobias, Dailyne D. Macasaet, and Elmer P. Dadios. A comparative analysis of machine learning algorithms modeled from machine vision-based lettuce growth stage classification in smart aquaponics. *International Journal of Environmental Science and Development*, 11(9):442–449, 2020.

[7] Feng Yu, Qian Zhang, Jun Xiao, Yuntao Ma, Ming Wang, Rupeng Luan, Xin Liu, Yang Ping, Ying Nie, Zhenyu Tao, and Hui Zhang. Progress in the application of cnn-based image classification and recognition in whole crop growth cycles. *Remote Sensing*, 15(12):2988, 2023.

[8] Yidong Wang, Mingge Wu, and Yunde Shen. Identifying the growth status of hydroponic lettuce based on yolo-efficientnet. *Plants*, 13(3):372, 2024.

[9] Robert G. de Luna, Elmer P. Dadios, Argel A. Bandala, and Ryan Rhay P. Vicerra. Tomato growth stage monitoring for smart farm using deep transfer learning with machine learning-based maturity grading. *AGRIVITA Journal of Agricultural Science*, 42(1):123–135, 2020.

[10] Glenn Jocher, Rizwan Munawar, et al. *Ultralytics YOLOv5 Architecture Summary*. Ultralytics Inc., September 2024. Version 8.3.19.

[11] Haiying Liu, Fengqian Sun, Jason Gu, and Lixia Deng. Sf-yolov5: A lightweight small object detection algorithm based on improved feature fusion mode. *Sensors*, 22(15):5817, 2022.

[12] Rahima Khanam and Muhammad Hussain. What is yolov5: A deep look into the internal features of the popular object detector. *arXiv*, 2024.

[13] Iason Katsamenis, Eleni Eirini Karolou, Agapi Davradou, and Dimitris Kalogeras. Tracon: A novel dataset for real-time traffic cones detection using deep learning. *Informatics*, 9(3):1–15, 2022.

[14] OpenGenus. Yolo v5 model architecture [explained]. *OpenGenus IQ*, n.d.

[15] Juan R. Terven and Diana M. Cordova-Esparaza. A comprehensive review of yolo: From yolov1 to yolov8 and beyond. *arXiv preprint arXiv:2304.00501*, 2023.