



Smart Farming: Improving Automated Broccoli Head Size Estimation

A Systematic Comparison of Image Preprocessing Techniques for YOLOv8n under Dutch Field Conditions

Huan Yi Ma¹

Supervisor(s): Cynthia Liem¹, Jeroen Wildenbeest², Surya Giri²

¹EEMCS, Delft University of Technology, The Netherlands

²Robotics, Inholland University of Applied Sciences, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 26, 2026

Name of the student: Huan Yi Ma

Final project course: CSE3000 Research Project

Thesis committee: Cynthia Liem¹, Jeroen Wildenbeest², Xucong Zhang¹

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Estimating the size of broccoli heads in the field is an important step towards automating broccoli harvesting, but it depends on first detecting the heads reliably and accurately in field imagery that contains harsh sunlight, camera blur, and leaf occlusion. This work investigates whether image preprocessing can improve such detection. Five common preprocessing techniques, CLAHE, unsharp masking, wavelet transform, median filtering, and bilateral filtering, were selected from the literature and evaluated individually and in five combinations. A systematic comparison was performed in which a YOLOv8n detector was trained and assessed with stratified five-fold cross-validation on a dataset of 394 field images. Performance was measured both over the entire dataset and over subsets representing difficult field conditions. On the full dataset the effect of preprocessing was negligible, as the baseline already performed close to the ceiling and differences between techniques were of the same order as the standard deviation between folds. However, in the difficult subsets preprocessing resulted in clear improvements, with the largest gains in the images on which the baseline struggled most. The most consistent results come from unsharp masking and the wavelet+unsharp combination. Combining techniques was not automatically beneficial. These results, obtained without significance testing, indicate that preprocessing is best understood as a tool for specific difficult conditions, rather than as a general improvement.

1 Introduction

In the Netherlands, broccoli is cultivated on roughly 2,500 hectares and is assessed and harvested mainly by hand during a season that runs from May to November [1]. Laborers visually assess each head and decide whether it has reached a marketable size. The process is labor-intensive, expensive, and difficult to sustain as the availability of seasonal labor declines [1]. There do exist commercially available mechanized harvesters that cut entire fields at once, but since individual plants do not grow uniformly, this can lead to significant crop waste [2]. To reduce these losses and move toward more sustainable precision agriculture, smart farming systems require accurate plant-level monitoring. Specifically, automated computer vision systems that can detect and accurately estimate the size of individual broccoli heads in the field to determine their optimal harvest time.

Recent work has shown that this is feasible. Mateijns [1] developed a predictive growth modeling pipeline for Dutch broccoli fields that retrains YOLOv8n [3], an object detection model, on field imagery obtained from Verdonk Broccoli and reached an F1-score of 0.958. Since the pipeline relies on head detection as the first step, any error introduced at detection affects the size estimation of the broccoli heads (Appendix A). However, the model must perform on raw

field imagery, which is often not ideal. The heads of broccoli sit low in the plant, causing the surrounding leaves to cast shadows and partially occlude the heads, while camera motion blur adds further noise. These conditions prove to be a challenge in broccoli head segmentation [4–7] and therefore it is important to improve detection accuracy under them. Outside of broccoli agriculture, applying image preprocessing before a deep-learning model has repeatedly been reported to improve detection and classification accuracy, in domains such as crop-disease recognition, biometric detection, and medical imaging [8–11], making it a promising option for the broccoli setting.

While the literature has explored many ways of improving the models for broccoli field conditions, relatively little work has studied whether improvements can be made by simply preprocessing the input image before the model sees it: Blok et al. [4] address the occlusion of broccoli heads by overlapping leaves with ORCNN, lowering the mean head-sizing error to 6.4 mm. Zuo et al. [5] present LBDC-YOLO for broccoli detection in complex field environments, reaching an average detection accuracy of 94.44%. Wang et al. [6] target varying lighting conditions with RSDF, which pairs YOLOv8n detection with SAM2 segmentation, reporting a detection accuracy of 96.9% and a head-diameter estimation accuracy of 90.31%. In these works, we observe that the reported improvements come from modifications to the models, not from changes to the input.

There are also works of outside broccoli agriculture, where the effects of image preprocessing on deep-learning performance is studied. Nouman Noor et al. [9] evaluate local contrast enhancement, wavelet transform, sigmoid and gamma correction, and median filtering as preprocessing techniques for YOLOv8 for crop-disease recognition. Persiya and Sasithradevi [10] report that combining CLAHE with YOLOv8 yields the highest accuracy for eye detection, with an mAP@0.50 of 0.995 and mAP@0.50-0.95 of 0.801. Nguyen-Tat et al. [11] pair 5 preprocessing combinations with five CNN architectures across eight medical image datasets and find that both the choice of preprocessing techniques and the choice of architecture can affect the accuracy significantly, with the Median-Mean hybrid filter and unsharp masking + bilateral filter being the most effective overall.

For broccoli specifically, no published work focuses on the effects of preprocessing on detection accuracy. Preprocessing is either left out, or applied without justification, or combined with architectural changes such that its individual contribution is hard to distinguish. The works of outside broccoli agriculture report clear gains from preprocessing alone, such as CLAHE improving YOLOv8 to an mAP@0.50 of 0.995 for eye detection [10] and unsharp masking with bilateral filtering ranking most effective across eight medical-imaging datasets [11]. Whether such gains carry over to broccoli head segmentation is unclear. Because the difficulties these techniques target, namely low contrast, sensor noise, and soft edges, are the same ones that characterize raw broccoli field imagery, partial crossover is expected. However, the effect is likely to be concentrated on the harder images, since a detector trained on this data should already perform

well on clean frames and therefore has little room to gain there. This thesis addresses this gap by evaluating common preprocessing techniques for broccoli head size estimation.

The main research question is as follows: *How can image preprocessing techniques improve automated broccoli head segmentation and size estimation in raw field imagery?* To answer it, the main question is divided into four research questions (RQ1–RQ4):

- RQ1** Which image preprocessing techniques are commonly applied to crop detection and segmentation tasks in computer vision literature?
- RQ2** How does applying image preprocessing techniques affect the performance of the model on the full dataset?
- RQ3** To what extent do image preprocessing techniques affect the performance on subsets of images characterized by difficult field conditions (e.g. harsh lighting, blur, occlusion)?
- RQ4** How do combinations of preprocessing techniques affect model performance compared to applying each technique individually?

This work makes three contributions. First, it reviews the preprocessing techniques most commonly applied to crop detection and segmentation and selects five that target the relevant field conditions: CLAHE, unsharp masking, wavelet denoising, median filtering, and bilateral filtering. Second, it systematically compares these techniques on raw broccoli field imagery with a YOLOv8n detector using stratified five-fold cross-validation, evaluating their effect both on the full dataset and on subsets characterized by harsh lighting, camera blur, and leaf occlusion. Third, it evaluates pipelines that combine the most promising individual techniques, testing whether stacking them yields further gains.

The remainder of the paper is structured as follows. Section 2 provides the methodology and evaluation metrics. Section 3 describes the preprocessing techniques used in the study. Section 4 describes the experimental setup and reports the results of the systematic comparison. Section 5 reflects on the ethical considerations and reproducibility of the work. Section 6 compares the findings in the context of the related work and discusses limitations. Section 7 concludes and suggests directions for further research.

2 Methodology

2.1 Selection of preprocessing techniques

RQ1 asks which image preprocessing techniques are most commonly applied to crop detection and segmentation tasks. Five techniques are selected for this study because they reoccur in the literature and target the field conditions identified in this work. Each technique is described in detail in Section 3, and only its role and reported effectiveness are mentioned here.

CLAHE (Contrast Limited Adaptive Histogram Equalization) enhances local contrast while limiting noise amplification, targeting harsh lighting and shadows that obscure the

head. It is one of the most widely used enhancement techniques and has been shown to improve both YOLO-based detection [10] and CNN-based image classification [8, 12].

Unsharp masking amplifies high-frequency details to restore edges softened by camera blur. It was among the most effective preprocessing techniques in a medical-imaging study [11].

Discrete wavelet transform denoising suppresses noise while preserving small details and is a well-established denoising approach that has been used in crop detection [9, 13], as well as medical imaging [11].

Median filtering removes impulsive noise while preserving edges. It is a standard denoiser whose effect on CNN performance has been studied directly [14] and has been applied as preprocessing for YOLOv8 crop-disease recognition [9].

Bilateral filtering smoothens noise in flat regions while preserving edges. It is a common denoiser evaluated for CNNs [14] and was identified as a strong preprocessing technique when combined with sharpening [11].

Because the individual techniques target different environmental conditions, five combinations are also evaluated to test whether their effectiveness can be improved (RQ4). In every combination, any contrast-enhancement or denoising stage is applied before sharpening, so that unsharp masking emphasizes actual edges rather than amplifying noise that an earlier stage would have otherwise removed.

Bilateral + unsharp and **wavelet + unsharp** combine edge-preserving denoising with sharpening, removing noise before the edges are emphasized. The former combination was reported to be effective in previous work [11]. **CLAHE + unsharp** combines contrast enhancement with sharpening, targeting weak contrast between head and background, and soft edges together. **Median + CLAHE** denoises before enhancing contrast, so that CLAHE does not amplify impulsive noise into the image. Finally, **bilateral + CLAHE + unsharp** chains all three stages, denoising, contrast enhancement, and sharpening, to target noise, illumination, and edge quality in a single pipeline.

Evaluating these combinations against the best individual techniques is what allows RQ4 to be answered. An example of every variant applied to an image can be found in Appendix B.

2.2 Dataset

The experiments use a dataset of 394 images containing broccoli heads from 6 measurement days from July 11 to July 23, 2025, each annotated with a bounding box. Although it has been derived from the Verdonk Broccoli field footage used by Mateijns [1], it is a distinct dataset. The images were extracted, annotated, and provided by a researcher at Inholland University of Applied Sciences. The footage is captured in field conditions and therefore contains the harsh lighting, camera blur, and leaf occlusion that motivate this study.

Due to the limited size of the dataset, performance is estimated with 5-fold cross-validation rather than a single held-out test. The folds are generated once using a single fixed random seed and are stratified across the measurement dates, so that every date is represented equally in each fold. The fold

assignment is reused for the baseline and for every preprocessed copy of the dataset, which guarantees that differences in the evaluation metrics are caused by preprocessing rather than from how the data is partitioned.

The dataset is used as provided and is not cleaned, as the focus of this work is on the relative effect of preprocessing rather than the absolute detection quality.

2.3 Detection model

All experiments use YOLOv8n [3], the smallest variant of YOLOv8. Following the same approach as Mateijnsen’s growth-modeling pipeline that this study extends [1], the network is initialized from weights pre-trained on the COCO dataset [15] and is then fine-tuned on the broccoli dataset described above. Choosing the smallest variant trades a lower accuracy ceiling for low computational cost and memory use, which makes it suitable for agricultural edge devices. A larger variant or segmentation-based detector could reach higher accuracy where computation is less constrained, but the model is deliberately held fixed here. The goal is not to find the best detector, but to isolate the effect of preprocessing.

2.4 Experimental design

The research questions are answered through a systematic comparison whose overall pipeline is shown in Figure 1. The pipeline generates, trains, and evaluates eleven dataset variants and compares each against the baseline, both over the full dataset and over the difficult subsets.

The starting point is the annotated dataset of 394 images. From it, the modular preprocessing stage (Section 3) produces eleven variants: the unmodified *baseline*, one variant for each of the five individual techniques, and one for each of the five combinations. Generating every variant from the same source images ensures that preprocessing is the only factor that differs between them.

Each variant is then partitioned with the same stratified 5-fold cross-validation, whose fold assignment is fixed by a single seed as described above, and a separate YOLOv8n model is trained on each variant using identical architecture and hyperparameters. Because the split and the training procedure are held constant across variants, any difference in the resulting metrics can be attributed to the preprocessing alone. Each trained model is evaluated on its held-out folds using precision, recall, mAP@0.50, and mAP@0.50-0.95.

The lower branch of Figure 1 defines the difficult subsets. The three difficult conditions considered, harsh lighting, camera blur, and leaf occlusion are taken from prior work on broccoli detection in the field (Section 1), where they are repeatedly reported as the main sources of difficulty [4–7]. To locate the images that exhibit these conditions, the predictions of the *baseline* are ranked by their IoU against the ground truth, and the worst-performing images are inspected manually. Each such image is assigned to the predefined condition it most clearly shows, while images that are difficult for several reasons at once, or for reasons outside the three conditions, are collected in a separate *mix* subset (Section 4).

Finally, every variant is compared to the baseline. The comparison is carried out over the entire dataset, addressing

RQ2, and over the difficult subsets, addressing RQ3. Comparing the two makes it possible to tell whether a change in the aggregate metrics reflects a genuine improvement on hard images or merely a shift on easy ones. The combinations are also compared to the best individual techniques to address RQ4.

2.5 Metrics

Detection performance and accuracy are reported using four standard object-detection metrics: precision, recall, mAP@0.50, and mAP@0.50-0.95. All four are built on the intersection over union (IoU), which is also used to rank the worst-performing images in the first step of the experimental design. Because the task contains a single object class (broccoli), the mean average precision reduces to the average precision of that class.

Intersection over Union (IoU). IoU measures the spatial overlap between a predicted bounding box B_p and a ground-truth box B_{gt} :

$$\text{IoU} = \frac{|B_p \cap B_{gt}|}{|B_p \cup B_{gt}|}, \quad (1)$$

where $|\dots|$ denotes the area of a region expressed as the number of pixels it contains. The numerator is the pixel area where the two boxes overlap and the denominator is the pixel area they cover together, so IoU is a dimensionless ratio ranging from 0 (no overlap) to 1 (perfect overlap). A prediction is counted as a true positive only when its IoU with a ground truth box is at least a matching threshold t (commonly 0.50).

Precision and Recall. With true positives (TP), false positives (FP), and false negatives (FN) determined by the IoU, precision and recall are defined as the following:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (2)$$

Precision captures how many of the detected heads are correct, which is relevant because false detections on background leaves would give false size estimates. Recall captures how many of the heads present are found, which matters because a missed head cannot be sized or harvested at the right time.

Average Precision and mAP. Both mAP metrics follow the COCO 101-point interpolated convention [15], which is the one implemented by the Ultralytics framework used here [3]. They are built from the precision and recall defined above in three steps.

First, every detection is labeled as a true or false positive at a fixed IoU threshold t : a predicted box is a true positive when its IoU with the ground truth head of the same class is at least t , and a false positive otherwise. The suffix $@t$ denotes this matching threshold (e.g. mAP@0.50 uses $t = 0.50$).

Second, a precision-recall curve $p(r)$ is created from the cumulative true and false positive counts. This curve is made monotonically non-increasing by replacing each value with the highest precision attained at any equal or greater recall:

$$p_{\text{interp}}(r) = \max_{\tilde{r} \geq r} p(\tilde{r}), \quad (3)$$

where $p_{\text{interp}}(r)$ is the interpolated (envelope) precision at recall r and $p(\tilde{r})$ is the raw precision at recall $\tilde{r}$.

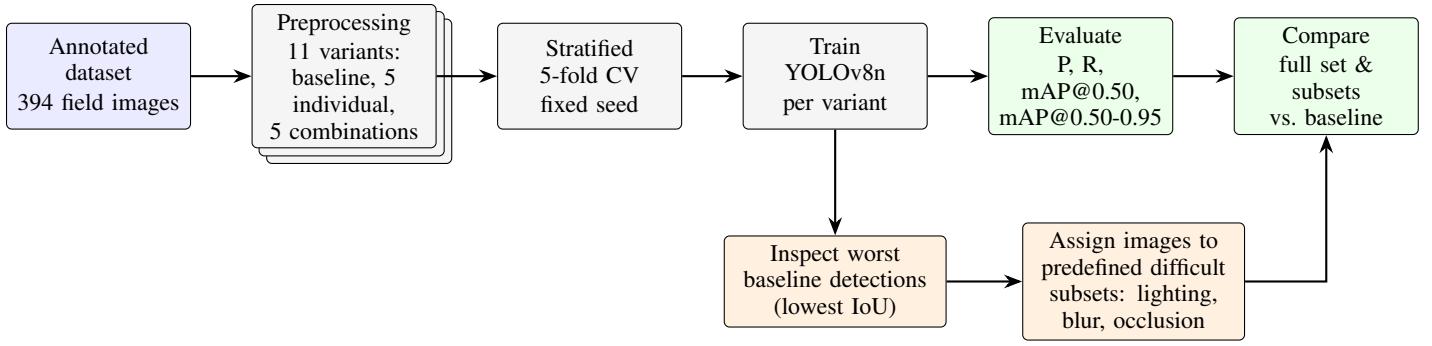


Figure 1: Overview of the experimental pipeline. The annotated dataset is transformed into eleven variants (the baseline, five individual preprocessing techniques, and five combinations). Each variant is split with stratified five-fold cross-validation, used to train a YOLOv8n model, and evaluated. The difficult conditions (harsh lighting, camera blur, leaf occlusion) are taken from prior work; the worst baseline detections, ranked by IoU, are inspected to select the images that exhibit them, forming the difficult subsets on which the variants are additionally compared.

Third, the average precision at threshold t is the area under this envelope, approximated by sampling it at 101 evenly spaced recall levels (the COCO 101-point rule) and integrating:

$$\text{AP}_t = \int_0^1 p_{\text{interp}}(r) dr \approx \frac{1}{101} \sum_{r \in \{0, 0.01, \dots, 1\}} p_{\text{interp}}(r), \quad (4)$$

where AP_t is the average precision obtained with matching threshold t .

The mean average precision averages the AP over all C classes, $\text{mAP}@t = \frac{1}{C} \sum_{c=1}^C \text{AP}_t^{(c)}$, where $\text{AP}_t^{(c)}$ is the average precision of class c . Because this task has a single class (broccoli), $C = 1$ and thus $\text{mAP}@t$ equals the AP of that class. Two variants are reported. **mAP@0.50** fixes $t = 0.50$ and rewards correct detection with loose localization. **mAP@0.50-0.95** repeats the matching at ten IoU thresholds and averages the results, and therefore rewards accurate localization:

$$\text{mAP}@0.50-0.95 = \frac{1}{10} \sum_{t \in \{0.50, 0.55, \dots, 0.95\}} \text{mAP}@t, \quad (5)$$

where the sum runs over the ten IoU thresholds from 0.50 to 0.95 in steps of 0.05. $\text{mAP}@0.50-0.95$ is the most informative metric for this study. Since the goal is to estimate the head *size*, the accuracy of the predicted bounding boxes matters as much as whether a head is detected at all, and only the stricter thresholds penalize loosely predicted boxes.

3 Image preprocessing techniques

This section describes in detail the five preprocessing techniques selected in Section 2 and the combinations formed from them. The reasoning for selecting these techniques and their reported effectiveness were given there. The focus here is on how each technique works and how it is applied to the dataset. Parameter values used in the experiments are stated for each technique. These are standard values commonly used for each technique, applied uniformly across the dataset rather than tuned individually.

3.1 Preprocessing pipeline

The preprocessing pipeline is implemented as a modular pipeline in which each technique is a stage that takes an image and returns a processed image of the same dimensions. A stage can be applied on its own or chained with others, making both individual technique and combination experiments possible. For each technique or combination, a fully processed copy of the dataset is generated once and stored, after which the YOLOv8n models are trained and evaluated on these copies. Because the techniques are applied to every image identically and the cross-validation folds are fixed (Section 2), the preprocessing is the only variable that differs between experiments. All techniques are applied to the image before it is resized to the network input resolution, so that they act on the full detail of the data.

3.2 CLAHE

Standard histogram equalization spreads the intensity histogram of an image across the full range to increase global contrast, using the cumulative distribution function (CDF) of the intensities as the mapping

$$T(r) = (L - 1) \text{CDF}(r), \quad (6)$$

where $T(r)$ is the output intensity to which an input intensity r is mapped, and L is the number of intensity levels. Applied globally, this over-amplifies contrast in locally uniform regions and amplifies noise. Adaptive histogram equalization instead computes a separate mapping for small non-overlapping tiles of the image, and CLAHE additionally clips each tile histogram at a *clip limit*, redistributing the clipped areas so that noise amplification is controlled [12]. To avoid visible tile boundaries, the per-tile mappings are combined by bilinear interpolation. In this study, CLAHE is applied to the luminance channel of the LAB color space, so that contrast is enhanced without shifting the colors that distinguish a head from the surrounding foliage. It is applied with a clip limit of 2.0 and a tile grid of 8×8 tiles.

3.3 Unsharp masking

Unsharp masking sharpens an image by adding a scaled copy of its own high-frequency content back onto it. A blurred

version of the image is first created with a Gaussian kernel, and the difference between the original and the blurred image (the high-frequency “mask”) is added back:

$$g(x, y) = f(x, y) + \lambda(f(x, y) - f_{\text{blur}}(x, y)), \quad (7)$$

where $g(x, y)$ is the sharpened output, $f(x, y)$ the input image, $f_{\text{blur}}(x, y)$ its Gaussian-blurred version, and λ the amount of sharpening. A larger λ produces strong edge emphasis at the cost of also amplifying noise, which is the reason why it is applied last in every combination (Section 2). It targets the soft head boundaries caused by camera blur. The Gaussian blur uses a standard deviation of $\sigma = 1.5$ and a sharpening amount of $\lambda = 1.0$.

3.4 Discrete wavelet transform

Wavelet denoising removes noise in the frequency domain while preserving spatial detail. A discrete wavelet transform (DWT) decomposes the image into one approximation subband and three detail subbands (horizontal, vertical, and diagonal) per level. This decomposition is repeated on the approximation subband for several levels. Noise is concentrated in the detail coefficients, so it is suppressed by thresholding those coefficients. Soft thresholding is used [16]:

$$\eta(w, T) = \text{sign}(w) \max(|w| - T, 0), \quad (8)$$

where $\eta(w, T)$ is the thresholded coefficient, w a detail coefficient, and T the threshold. The image is then reconstructed with the inverse DWT. Because the approximation subband does not get changed, the overall structure and edges are preserved while noise is removed. The decomposition uses Daubechies-4 (db4) wavelet [17] over 2 levels. Rather than a fixed threshold, T is estimated per channel from the detail subband using Donoho’s median absolute deviation (MAD) estimator [18]:

$$\hat{\sigma} = \frac{\text{median}(|w|)}{0.6745}, \quad T = \hat{\sigma} \sqrt{2 \ln N}, \quad (9)$$

where $\hat{\sigma}$ is the estimated noise standard deviation, the constant 0.6745 scales the median absolute deviation to that estimate under a Gaussian noise assumption, T is the resulting threshold, and N is the number of pixels in the channel. This universal threshold adapts to the noise level present in each image rather than applying a fixed value across the dataset.

3.5 Median filtering

Median filtering is a non-linear filter that replaces each pixel with the median intensity of the pixels in a square window centered on it:

$$g(x, y) = \text{median}\{f(i, j) : (i, j) \in W_{x, y}\}, \quad (10)$$

where $g(x, y)$ is the filtered output, $f(i, j)$ the input intensities, and $W_{x, y}$ the window centered on (x, y) . The median filter is considered to be the most effective way to remove impulsive (salt and pepper) noise [14]. A window $W_{x, y}$ of 5×5 pixels was used here.

3.6 Bilateral filtering

Bilateral filtering smoothens an image while preserving its edges by averaging each pixel only with neighbors that are close in both space and intensity. The output at pixel p is:

$$\text{BF}[I]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) G_{\sigma_r}(|I_p - I_q|) I_q, \quad (11)$$

where $\text{BF}[I]_p$ is the filtered value at pixel p , I_q the intensity of a neighboring pixel q in the window S , I_p the intensity at p , G_{σ_s} and G_{σ_r} are Gaussian functions of the spatial distance and the intensity difference respectively, and W_p is the normalization factor that makes the weights sum to one. The range term G_{σ_r} gives a low weight to neighbors whose intensity differs strongly, which prevents the smoothing of edges. The filter diameter is set to $d = 9$ pixels, and the spatial and range parameters are set to $\sigma_s = 50$ and $\sigma_r = 50$.

3.7 Combinations

Five combinations are evaluated: bilateral + unsharp, CLAHE + unsharp, median + CLAHE, wavelet + unsharp, and bilateral + CLAHE + unsharp. Each combination is realized by passing an image through the corresponding stages of the pipeline in sequence. As motivated in Section 2, any denoising or contrast-enhancement stage is applied before unsharp masking, so that sharpening emphasizes real edges rather than re-amplifying noise that an earlier stage removed. The same parameter values listed above for the individual techniques are reused unchanged within the combinations, so that any difference in performance reflects the interaction between the techniques.

4 Experimental Setup and Results

4.1 Experimental setup

All experiments were run on a single workstation equipped with an NVIDIA GeForce RTX-5080 (16GB GDDR7) on Windows 11. The used software was Python 3.12 with PyTorch 2.12 (CUDA 12.8) and Ultralytics 8.4.51 [3]. All training and testing used the GPU.

The YOLOv8n model is initialized from the COCO-pretrained `yolo8n.pt` weights and fine-tuned on the brocoli data. The same is done for every preprocessing variant so that the input transformation is the only factor that changes between experiments. Each model is trained for up to 100 epochs with early stopping (patience 30), a batch size of 16, and an input resolution of 1280×1280 pixels. Training data augmentation uses the Ultralytics defaults: HSV jitter ($hsv_h = 0.015$, $hsv_s = 0.7$, $hsv_v = 0.4$), random scaling ($scale = 0.5$), translation ($translate = 0.1$), and mosaic augmentation ($mosaic = 1.0$). A fixed random seed (42) is used throughout the experiments so that the data splits and weight initialization are the same across all variants.

To answer RQ3, four subsets of difficult images are evaluated separately. The three difficult conditions, *camera blur*, *harsh lighting*, and *leaf occlusion*, are the ones identified in the literature (Section 1). The images characterizing them were selected from the baseline’s worst detections (Appendix C), as described in Section 2. The fourth subset *mix*

contains difficult images that did not clearly belong to one of the three conditions or that combine several of them. Each subset contains 31-32 images and between 36 and 44 annotated broccoli heads (camera blur: 36, harsh lighting: 39, leaf occlusion: 44, mix: 40). Because the cross-validation assigns each image to exactly one validation fold, every subset image is scored by the model that never saw it during training. The resulting predictions are gathered and scored once with the same metrics as the overall dataset (precision, recall, mAP@0.50, mAP@0.50-0.95).

4.2 Results on the full dataset

Table 1 reports the cross-validated performance of every technique and combination on the full dataset, as the mean and standard deviation over the five folds. The differences between variants are small: all eleven variants are within roughly one percentage of each other on mAP@0.50 and withing half a percentage on mAP@0.50-0.95, and the standard deviations are of the same order as the differences between variants. This shows that on the whole dataset, no single preprocessing variant produces a decisive change in overall detection accuracy compared to the baseline.

However, a consistent pattern does show in the results. Unsharp masking and the wavelet+unsharp combination give the best mAP@0.50 (0.988 and 0.989), and wavelet+unsharp ties CLAHE for the best mAP@0.50-0.95 (0.751), slightly above the baseline (0.748). Another observation is that the baseline has the highest precision while the preprocessing variants tend to have a higher recall. This shows that preprocessing makes the model detect more heads at a small cost in precision.

4.3 Results on the difficult-condition subsets

Tables 2–5 report the performance of the preprocessing variants on the blurry, harsh lighting, leaf occlusion and mix subsets respectively. Here the differences between the baseline and the preprocessed variants are much larger.

On the *blurry* subset (Table 2) the wavelet+unsharp combination seems to be the overall best performing, having perfect precision and recall and an mAP@0.50 of 0.995, which is well above the baseline 0.973. This result can be expected, since wavelet denoising followed by sharpening targets the soft edges produced by camera blur.

The *harsh lighting* subset (Table 3) seems to be the easiest condition overall. Precision is 1.0 for most variants including the baseline, while CLAHE, median filtering, wavelet+unsharp, and bilateral+CLAHE+unsharp increase the recall to 1.0 and mAP@0.50 to 0.995. Out of these four, median and wavelet+unsharp increase the mAP@0.50-0.95 the most to 0.729 and 0.731 respectively, making them the overall best performing techniques on this subset.

On the *leaf occlusion* subset (Table 4) wavelet+unsharp again gives the best gives the best mAP@0.50 (0.993), and the techniques that preserve edges (bilateral, wavelet) give the best mAP@0.50-0.95.

The *mix* subset (Table 5) is the one with the biggest differences in performance. Here the baseline has the lowest recall of 0.825, an mAP@0.50 of 0.844, and an mAP@0.50-0.95 of 0.624, which is by far its worst performance on any of the

subsets. On the other hand, unsharp masking increases both the recall and mAP@0.50 to 0.971 and the mAP@0.50-0.95 to 0.702, while median+CLAHE has the highest recall of 1.0 and mAP@0.50-0.95 of 0.705. These two variants seem to be performing the best on this subset and every variant seems to improve on recall, mAP@0.50, and mAP@0.50-0.95.

Across the four subsets, unsharp masking and its combinations (particularly wavelet+unsharp) seem to be the most consistent performers, whereas no variant is globally best on every metric on every subset.

5 Responsible Research

This section reflects on the ethical aspects of the research and on the reproducibility of the methods, covering the source and handling of the data, the limitations of relying on a single dataset, the societal impact of the work on agricultural labor, and the use of large language models during the project.

5.1 Data source and management

The dataset used in this study was provided by Inholland University of Applied Sciences and is not publicly available. It was extracted from footage of a field at Verdonk Broccoli and annotated by a researcher at Inholland. The imagery contains only broccoli plants in an agricultural setting and does not contain identifiable people or other personally identifiable information, thus the privacy risk associated with handling it is low. Nevertheless, because the data was shared for the purpose of this research and remains the property of the providing parties, it has been stored securely, used only for the experiments described here, and not redistributed.

The confidentiality of the dataset has a direct consequence for reproducibility: the exact results cannot be reproduced by a third party without access to the same data. To compensate, the methodology has been described in enough detail to be reproduced on comparable data, including the used hardware and software, the model and its hyperparameters, the data partitioning and fixed seed, the preprocessing techniques with their parameters, and the evaluation metrics. In addition, the full code used to generate the preprocessed variants and to train and evaluate the models is publicly available at <https://github.com/HuanYiMa/ResearchProject>.

5.2 Impact on agricultural labor

Improving automated crop detection and estimation contributes to the automation of agriculture, which has a double-edged effect on farm labor. On one hand, it supports systems that can reduce dependence on manual harvesting, work that is physically demanding and affected by labor shortages. On the other hand, wider deployment of such systems can eventually replace agricultural jobs, with consequences for laborers that depend on them. Whether the resulting impact is beneficial depends on how downstream systems are designed and on the social and economic choices made around their adoption.

5.3 Use of large language models

Large language models (LLMs) were used as an assisting tool during this project, both for writing and for implementation.

Table 1: Cross-validated detection performance on the full dataset (mean $\pm$ standard deviation over five folds). The best mean for each metric is shown in bold. No variant changes overall accuracy decisively, as all eleven stay within roughly one percentage point on mAP@0.50 and the difference is comparable to the per-fold standard deviation.

Variant	Precision	Recall	mAP@0.50	mAP@0.50-0.95
Raw (baseline)	0.969 $\pm$ 0.006	0.963 $\pm$ 0.016	0.980 $\pm$ 0.007	0.748 $\pm$ 0.012
CLAHE	0.967 $\pm$ 0.015	0.974 $\pm$ 0.014	0.986 $\pm$ 0.006	0.751 $\pm$ 0.019
Bilateral	0.963 $\pm$ 0.012	0.959 $\pm$ 0.012	0.984 $\pm$ 0.007	0.748 $\pm$ 0.015
Unsharp	0.964 $\pm$ 0.015	0.976 $\pm$ 0.008	0.988 $\pm$ 0.004	0.749 $\pm$ 0.016
Wavelet	0.964 $\pm$ 0.007	0.965 $\pm$ 0.012	0.979 $\pm$ 0.009	0.746 $\pm$ 0.016
Median	0.968 $\pm$ 0.014	0.969 $\pm$ 0.014	0.987 $\pm$ 0.005	0.745 $\pm$ 0.011
Bilateral + Unsharp	0.956 $\pm$ 0.032	0.967 $\pm$ 0.014	0.982 $\pm$ 0.008	0.746 $\pm$ 0.014
CLAHE + Unsharp	0.951 $\pm$ 0.017	0.969 $\pm$ 0.018	0.983 $\pm$ 0.009	0.746 $\pm$ 0.012
Median + CLAHE	0.966 $\pm$ 0.016	0.966 $\pm$ 0.019	0.986 $\pm$ 0.003	0.747 $\pm$ 0.011
Wavelet + Unsharp	0.963 $\pm$ 0.018	0.972 $\pm$ 0.023	0.989 $\pm$ 0.001	0.751 $\pm$ 0.019
Bilateral + CLAHE + Unsharp	0.957 $\pm$ 0.017	0.957 $\pm$ 0.021	0.980 $\pm$ 0.009	0.746 $\pm$ 0.014

Table 2: Performance on the *blurry* subset (32 images, 36 broccoli heads). Best per metric in bold. Wavelet+unsharp is the strongest variant, reaching perfect precision and recall and the highest mAP@0.50 (0.995 vs. 0.973 for the baseline).

Variant	Precision	Recall	mAP@0.50	mAP@0.50-0.95
Raw (baseline)	0.972	0.972	0.973	0.697
CLAHE	1.000	0.917	0.915	0.719
Bilateral	0.998	0.944	0.945	0.688
Unsharp	0.994	0.944	0.974	0.725
Wavelet	0.968	0.944	0.944	0.722
Median	0.998	0.944	0.945	0.727
Bilateral + Unsharp	1.000	0.944	0.945	0.721
CLAHE + Unsharp	0.993	0.972	0.975	0.711
Median + CLAHE	0.998	0.944	0.945	0.745
Wavelet + Unsharp	1.000	1.000	0.995	0.716
Bilateral + CLAHE + Unsharp	0.996	0.972	0.975	0.718

Table 3: Performance on the *harsh-lighting* subset (31 images, 39 broccoli heads). Best per metric in bold. The easiest condition overall, with precision at or near 1.0 for almost every variant, including the baseline.

Variant	Precision	Recall	mAP@0.50	mAP@0.50-0.95
Raw (baseline)	1.000	0.949	0.945	0.685
CLAHE	1.000	1.000	0.995	0.688
Bilateral	1.000	0.974	0.975	0.714
Unsharp	0.970	0.974	0.973	0.707
Wavelet	1.000	0.970	0.975	0.729
Median	1.000	1.000	0.995	0.729
Bilateral + Unsharp	0.974	0.949	0.937	0.691
CLAHE + Unsharp	0.927	0.974	0.970	0.666
Median + CLAHE	1.000	0.923	0.925	0.638
Wavelet + Unsharp	1.000	1.000	0.995	0.731
Bilateral + CLAHE + Unsharp	1.000	1.000	0.995	0.707

In writing, they were used to improve the phrasing, formatting, and clarity of the text that originates from the author’s own work and findings. In implementation, they were used to help with the implementation and debugging of the preprocessing and experiment code. All LLM output was reviewed by the author: generated text was checked against the original work and cited sources, and generated code was checked and tested to confirm it behaved as intended. The research questions, experimental design, interpretation of the results, and conclusions are the author’s own and the author takes full responsibility for the content of this paper.

Table 4: Performance on the *leaf-occlusion* subset (32 images, 44 broccoli heads). Best per metric in bold. Wavelet+unsharp gives the highest mAP@0.50 (0.993), while the edge-preserving filters bilateral and wavelet give the best localization (mAP@0.50-0.95 0.717 and 0.714).

Variant	Precision	Recall	mAP@0.50	mAP@0.50-0.95
Raw (baseline)	0.997	0.977	0.975	0.655
CLAHE	1.000	0.952	0.955	0.698
Bilateral	0.956	0.977	0.972	0.717
Unsharp	0.995	0.955	0.955	0.675
Wavelet	0.976	0.929	0.949	0.714
Median	0.996	0.909	0.934	0.660
Bilateral + Unsharp	0.910	0.922	0.929	0.661
CLAHE + Unsharp	0.933	0.953	0.965	0.664
Median + CLAHE	0.974	0.909	0.905	0.640
Wavelet + Unsharp	0.994	0.955	0.993	0.670
Bilateral + CLAHE + Unsharp	1.000	0.954	0.955	0.700

6 Discussion

6.1 Interpretation of results

The following caution applies to all of the below. No significance testing has been performed. On the full dataset the standard deviation across fold is of the same order as the differences between variants, so those differences cannot be separated from noise. On the subsets, each contains only 36-44 annotated heads, so a single missed or false detection moves the metrics substantially. The patterns described here are therefore observations and possible explanations rather than statistically established claims.

The main finding is that preprocessing has almost no effect on the metrics of the overall dataset, but a visible effect on the difficult subsets. Two factors explain the results on the overall dataset. First, the baseline results are already close to the ceiling in this dataset, with precision and recall around 0.97 and an mAP@0.50 of 0.98, leaving little room for improvement. Second, the large majority of the dataset contains images that do not have difficult conditions, so an improvement on difficult images is diluted when it is averaged over the full dataset. The difficult subsets remove this dilution, which is why the differences become visible there.

Within the subsets a few consistent patterns can be observed. Preprocessing tends to raise the recall while slightly lowering precision compared to the baseline. This indicates

Table 5: Performance on the *mix* subset (32 images, 40 broccoli heads). Best per metric in bold. The hardest subset, where the baseline is weakest (recall 0.825) and every variant improves on it, led by unsharp masking and median+CLAHE.

Variant	Precision	Recall	mAP@0.50	mAP@0.50-0.95
Raw (baseline)	0.972	0.825	0.844	0.624
CLAHE	0.927	0.950	0.936	0.697
Bilateral	0.921	0.875	0.869	0.659
Unsharp	0.951	0.971	0.971	0.702
Wavelet	0.972	0.875	0.870	0.649
Median	0.923	0.950	0.933	0.682
Bilateral + Unsharp	0.902	0.924	0.915	0.658
CLAHE + Unsharp	0.969	0.950	0.942	0.683
Median + CLAHE	1.000	0.933	0.945	0.705
Wavelet + Unsharp	0.881	0.925	0.912	0.703
Bilateral + CLAHE + Unsharp	0.920	0.864	0.858	0.651

that the model detects more heads at the cost of more false positives. For the downstream goal of size estimation, where a head that is not detected cannot be sized at all, trading precision for recall is generally favorable, given that the additional false positives can be filtered later. The techniques that target edges, unsharp masking and especially wavelet+unsharp, are the most consistent performers, and their advantage is the clearest on the blurry subset. This matches the underlying mechanism, as restoring the soft head boundary produced by camera blur is exactly what sharpening after edge preserving denoising is intended to do.

The effect is also concentrated on the metrics that matter the most for size estimation. mAP@0.50 is already saturated for almost every variant, so the more relevant change is in recall and in mAP@0.50-0.95, the metric that rewards tight localization. The largest gains appear on the *mix* subset where the baseline is weakest (recall 0.825, mAP@0.50-0.95 0.624) and where every variant improves on it, unsharp masking and median+CLAHE especially.

For the combinations (RQ4), the results do not support the assumption that stacking techniques is automatically beneficial. Only wavelet+unsharp consistently matches or exceeds its strongest individual component. Several combinations, such as bilateral+CLAHE+unsharp, perform at or below the level of their individual techniques on the occlusion and mix subsets. This suggests that combining techniques is useful only when the stages target a specific goal while complementing each other, such as edge-preserving denoising followed by sharpening, and that adding further stages risks reducing their effects.

6.2 Comparison with related work

The detection quality observed here is broadly in line with the work this study builds on. Using an 80/20 train/test split on imagery from the same Verdonk Broccoli footage, Mateijsen [1] reports a precision of 0.931, a recall of 0.987, an F1-score of 0.958, an mAP@0.50 of 0.988, and an mAP@0.50-0.95 of 0.808. The mAP@0.50 obtained here is effectively identical (baseline 0.980), which shows that detection at an IoU threshold of 0.5 is just as reliable on the current annotated dataset. The F1 is also comparable: the baseline precision and recall of 0.97 and 0.96 correspond to an F1-score of roughly 0.966. However, the baseline here leans more towards precision while Mateijsen’s model leans towards recall.

Interestingly, the preprocessing variants improve the recall at a small cost in precision and thus move the balance closer to Mateijsen’s results.

There is a clear gap in the mAP@0.50-0.95, where this study reaches about 0.75 against Mateijsen’s 0.808. Because the mAP@0.50 values are pretty much equal, the difference in the mAP@0.50-0.95 is most likely due to how accurately the predicted boxes match the ground truth rather than to whether heads are detected at all. This behavior is expected from the annotation inaccuracies in the current dataset: the missing and slightly inaccurate ground truth boxes (see 6.3), which were left uncleaned, reduce the achievable overlap and therefore decreases the mAP@0.50-0.95 specifically.

Compared with preprocessing studies outside broccoli agriculture, the picture is more nuanced. Nouman Noor et al. [9] for crop-disease recognition, Persiya and Sasithradevi [10] for eye detection, and Nguyen-Tat et al. [11] for medical imaging all report clear gains from preprocessing, whereas here the effect on the full dataset is negligible. The most likely reason is headroom: those baselines had more room to improve, while the baseline here is already near the ceiling, and preprocessing only yields comparable gains once the difficult subsets remove that ceiling effect. Regarding preprocessing techniques, there is partial agreement: Nguyen-Tat et al. found sharpening combined with denoising (unsharp masking with bilateral filtering) among the most effective methods, which matches the consistent performance of unsharp masking and wavelet+unsharp on the difficult subsets here. However, the improvement from CLAHE that Persiya and Sasithradevi reported for eye detection did not transfer, as CLAHE was not among the most consistent techniques in this study, suggesting that the benefit of a given technique depends on the task and the conditions it targets.

6.3 Limitations

Several limitations arise with these findings. The stated research goal concerns broccoli head segmentation and size estimation, but the experiments evaluate bounding box detection only, so whether the results also carry over to segmentation is something that cannot be said. Each technique was applied with a single fixed parameter setting, and those settings were reused inside the combinations. Different parameters could result in different outcomes, and no parameter tuning was performed. Only five of the many possible preprocessing combinations were evaluated, so no claims can be made that the resulting pipelines are optimal. All results were obtained with a single model, YOLOv8n, on one dataset of 394 images, which limits the extent to which they generalize to other architectures and data.

The subset analysis has additional weaknesses. Each subset contains only 36-44 annotated heads, which means a single missed or false detection can shift the metrics considerably. The subsets were created by subjective visual judgment of someone who is not an expert, which may introduce inconsistency in how the images were assigned to subsets. Finally, detection accuracy is limited by the quality of the dataset annotations. Some heads in multi-head images are missing annotations, and some bounding boxes are slightly inaccurate. This lowers the metrics, and on the small subsets specifically,

may interact with individual techniques in ways that are difficult to identify. Together with the absence of significance testing above, these limitations mean that the results are best read as indicative evidence about where and how preprocessing helps, rather than as definitive measurements.

7 Conclusions and Future Work

7.1 Conclusion

This study investigated how image preprocessing can improve automated detection of broccoli heads in raw field imagery, the first step of a pipeline that estimates head size to determine when to harvest. The main question was how image preprocessing techniques can optimize automated broccoli head segmentation and size estimation in raw field imagery. It was answered through a systematic comparison in which a YOLOv8n detector was trained and evaluated, using stratified five-fold cross-validation, on a baseline dataset and copies transformed by individual preprocessing techniques and combinations of them.

RQ1 asked which preprocessing techniques are commonly applied to crop detection and segmentation. A review of the literature identified five reoccurring techniques: CLAHE, unsharp masking, wavelet denoising, median filtering, and bilateral filtering.

RQ2 was about the effect of preprocessing on overall performance. Here, the effect was found to be negligible: all variants stayed within roughly one percent of the baseline on mAP@0.50 and within half a percent on mAP@0.50-0.95. These differences were of the same order as the variation between cross-validation folds, so they cannot be separated from noise. The baseline detector already performed close to the ceiling on the full dataset, leaving little room for any preprocessing technique to make a substantial difference when averaged over all images. The main observable trend was a small increase in recall with a small decrease in precision.

RQ3 asked about performance on subsets of images characterized by difficult field conditions. On these subsets the differences were considerably larger. Preprocessing led to clear improvements over the baseline, with the largest gains showing on the hardest images. The techniques that sharpen edges, unsharp masking and the wavelet+unsharp combination in particular, were the most consistent performers. Their advantage was clearest on blurry images, matching their intended purpose of restoring soft head boundaries.

RQ4 asked how combinations compare to individual techniques. Combining techniques was not found to be automatically beneficial. Only wavelet+unsharp consistently matched or exceeded its strongest individual component, while several combinations performed equal or worse than their individual techniques. This shows that combinations help only when their stages target a specific goal while complementing each other, and that adding more stages can reduce their effectiveness.

Together, these findings suggest that preprocessing is best understood as a tool for specific difficult field conditions rather than as a general improvement for detection. Its benefit is not visible in the metrics of the full dataset, but becomes noticeable on the subsets of hard images where the baseline

struggles. It's effectiveness is mainly in detecting heads that would otherwise be missed and by improving localization, both of which are valuable for the downstream goal of head size estimation. These conclusions are drawn from a single model and dataset and without significance testing, so they should be read as indicative evidence about where preprocessing helps rather than as definitive measurements.

7.2 Future Work

Several directions follow from this study. Because the observed differences are small relative to the standard deviation between folds and the difficult subsets are small, significance testing is needed to verify which of the observed effects are real. The second direction is parameter tuning, as every technique was applied with a single fixed setting. Tuning these parameters and trying out more combinations could result in more effective pipelines. An extension to this idea is to apply machine learning to the preprocessing pipeline to select or finetune techniques, similar to the work done by Campos et. al [12].

The evaluation itself could also be extended. The experiments used only YOLOv8n and measured bounding box detection, whereas the stated goal talks about segmentation and size estimation. Repeating the study with other detection and segmentation models, and measuring the effect of preprocessing on the final diameter estimate rather than on detection alone, would test whether the benefits found here carry through the entire pipeline. Finally, validating the results on larger, more diverse, and more cleanly annotated datasets would clarify how far they generalize beyond the single field and the annotation quality of the dataset used here.

References

- [1] N. Mateijnsen, "Field-Based Predictive Growth Modeling of Broccoli (*Brassica oleracea* var. *italica*) within Precision Agriculture: Understanding Field Growth Dynamics for Data-Driven Agriculture," Ph.D. dissertation, Delft University of Technology, Delft, Dec. 2025. [Online]. Available: <https://repository.tudelft.nl/record/uuid:2bb6b9b4-c552-4286-ac5d-676bb0db9122>
- [2] P. M. Blok, R. Barth, and W. van den Berg, "Machine vision for a selective broccoli harvesting robot," *IFAC-PapersOnLine*, vol. 49, no. 16, pp. 66–71, Jan. 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896316315749>
- [3] G. Jocher, J. Qiu, M. Liu, S. Lyu, F. C. Akyon, and M. E. Kalfaoglu, "Ultralytics YOLO26: Unified Real-Time End-to-End Vision Models," 2026, original-date: 2022-09-11T16:39:45Z. [Online]. Available: <https://arxiv.org/abs/2606.03748>
- [4] P. M. Blok, E. J. van Henten, F. K. van Evert, and G. Kootstra, "Image-based size estimation of broccoli heads under varying degrees of occlusion," *Biosystems Engineering*, vol. 208, pp. 213–233, Aug. 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1537511021001203>

- [5] Z. Zuo, S. Gao, H. Peng, Y. Xue, L. Han, G. Ma, and H. Mao, "Lightweight Detection of Broccoli Heads in Complex Field Environments Based on LBDC-YOLO," *Agronomy*, vol. 14, p. 2359, Oct. 2024.
- [6] Z. Wang, J. Xu, X. Zhang, A. Shen, Z. Zhang, and Y. Xun, "In Situ Detection and Measurement of Broccoli Heads Under Different Lighting Conditions Using Proximal Remote Sensing," *Journal of Field Robotics*, vol. 42, no. 8, pp. 4101–4114, 2025, eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.70008>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.70008>
- [7] C. M. Badgujar, A. Poullose, and H. Gan, "Agricultural object detection with You Only Look Once (YOLO) Algorithm: A bibliometric and systematic literature review," *Computers and Electronics in Agriculture*, vol. 223, p. 109090, Aug. 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169924004812>
- [8] M. J. Yousif, "Enhancing The Accuracy of Image Classification Using Deep Learning and Preprocessing Methods," *Artificial Intelligence & Robotics Development Journal*, 2023. [Online]. Available: <https://testojs.acaa-p.com/index.php/AIRDJ/article/view/8>
- [9] M. Nouman Noor, M. Masab, F. Haneef, M. Hussain, M. Yaqoob, T. Mazhar, M. Amir Khan, and G. Aldehim, "An Effective Approach for Recognition of Crop Diseases Using Advanced Image Processing and YOLOv8," *Food Science & Nutrition*, vol. 14, no. 2, p. e71504, Feb. 2026. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC12887443/>
- [10] P. J. and S. A., "Synergistic fusion: An integrated pipeline of CLAHE, YOLO models, and advanced super-resolution for enhanced thermal eye detection," *PLOS One*, vol. 20, no. 7, p. e0328227, Jul. 2025. [Online]. Available: <https://pmc.ncbi.nlm.nih.gov/articles/PMC12273955/>
- [11] T. B. Nguyen-Tat, T. Q. Hung, P. T. Nam, and V. M. Ngo, "Evaluating pre-processing and deep learning methods in medical imaging: Combined effectiveness across multiple modalities," *Alexandria Engineering Journal*, vol. 119, pp. 558–586, Apr. 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1110016825001176>
- [12] G. F. C. Campos, S. M. Mastelini, G. J. Aguiar, R. G. Mantovani, L. F. d. Melo, and S. Barbon, "Machine learning hyperparameter selection for Contrast Limited Adaptive Histogram Equalization," *EURASIP Journal on Image and Video Processing*, vol. 2019, no. 1, p. 59, May 2019. [Online]. Available: <https://doi.org/10.1186/s13640-019-0445-4>
- [13] C. Akdoğan, T. Özer, and Y. Oğuz, "PP-YOLO: Deep learning based detection model to detect apple and cherry trees in orchard based on Histogram and Wavelet preprocessing techniques," *Computers and Electronics in Agriculture*, vol. 232, p. 110052, May 2025. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169925001589>
- [14] R. Kaur, G. Karmakar, and M. Imran, "Impact of Traditional and Embedded Image Denoising on CNN-Based Deep Learning," *Applied Sciences*, vol. 13, no. 20, p. 11560, Jan. 2023. [Online]. Available: <https://www.mdpi.com/2076-3417/13/20/11560>
- [15] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.
- [16] D. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, May 1995. [Online]. Available: <http://ieeexplore.ieee.org/abstract/document/382009>
- [17] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 41, no. 7, pp. 909–996, 1988, eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.3160410705>. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160410705>
- [18] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, Sep. 1994. [Online]. Available: <https://doi.org/10.1093/biomet/81.3.425>

A Head Diameter Estimation from Bounding Boxes

In the growth-modeling pipeline this work extends, the physical head diameter is not measured directly but derived from the detection's bounding box. Mateijsen [1] estimates it as

$$D = \frac{D_{px} H_1 (f_x + D_{px} C_b)}{f_x^2}, \quad (12)$$

where D is the estimated head diameter, D_{px} the diameter in pixels (taken as the longest side of the bounding box), H_1 the median depth over the central region of the box, f_x the focal length of the camera, and C_b an empirically determined correction ratio ($C_b = 0.1$). Expanding the product shows that D contains a term in D_{px}^2 , so an inaccurate box therefore propagates directly, and amplified, into the size estimate.

B Example of each preprocessing variant

Figure 2 shows the same broccoli image after each of the ten preprocessing variants.

C Example predictions on the difficult condition subsets

Figure 3 shows four representative broccoli head predictions for each difficult condition subset. Predicted bounding boxes are drawn in red and the ground-truth boxes in green.

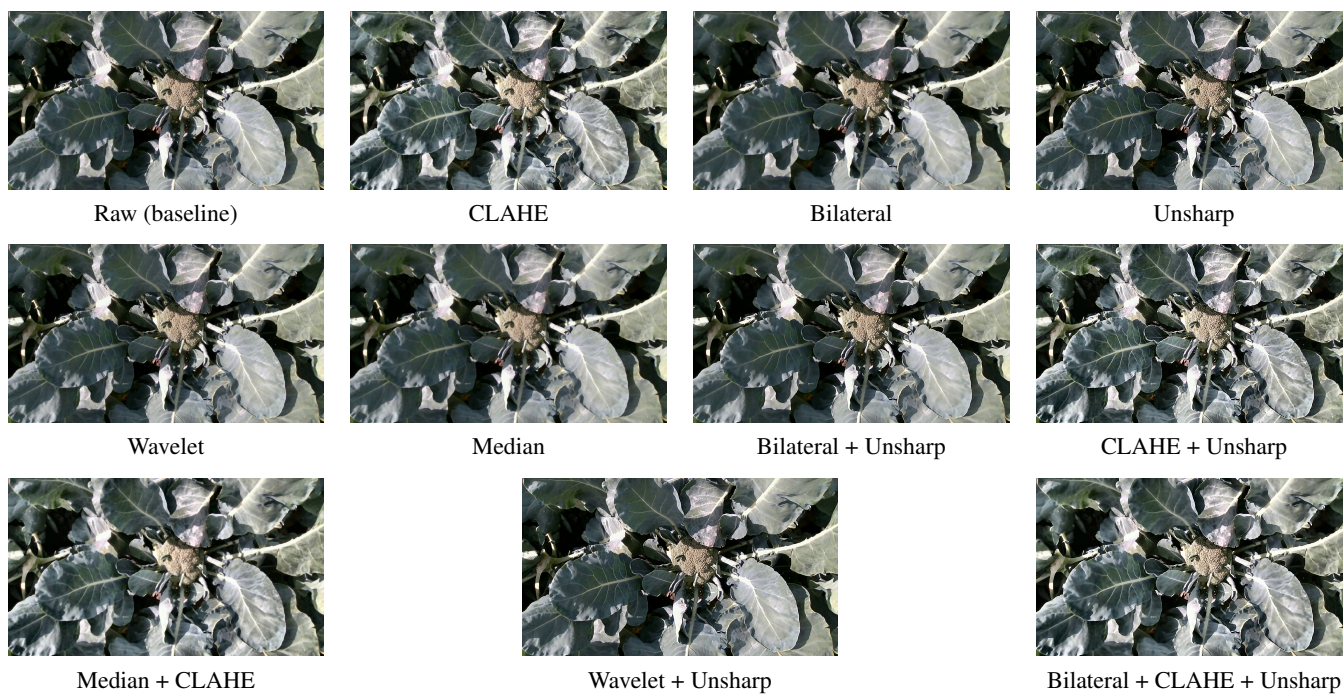
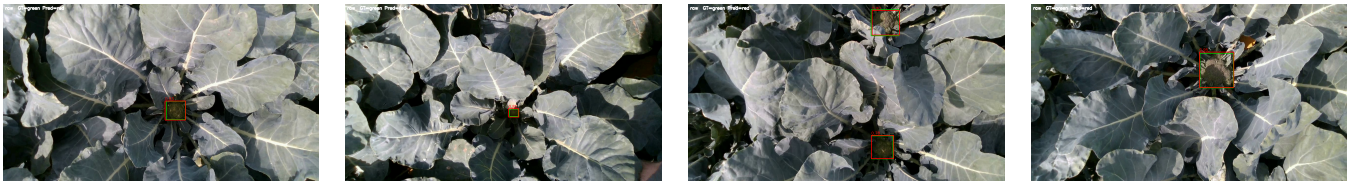


Figure 2: The same broccoli image after each preprocessing variant. The baseline is the unmodified image. The remaining panels show the five individual techniques and five combinations applied to it.

Blurry



Harsh lighting



Leaf occlusion



Mix

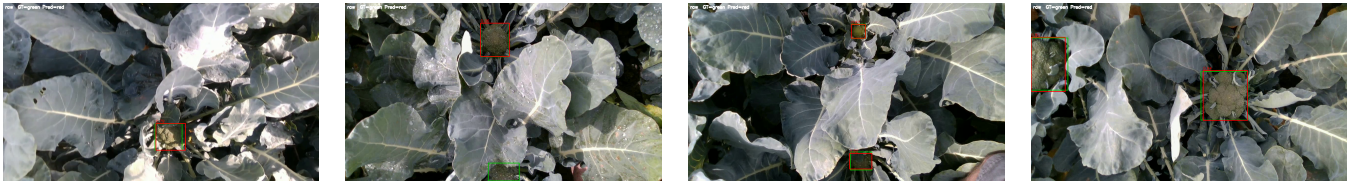


Figure 3: Example broccoli head predictions on the four difficult subsets. Each row shows four representative images from one subset. Predicted boxes are red and ground truth boxes green.