Mask R-CNN と SSIM に基づくマルチスペクトル画像処理システムによるピーマンの 果実検出

Multispectral Imaging System Fruit Detection for Green Pepper based on Mask R-CNN and SSIM

知能機械工学コース

機械・航空システム制御研究室　1247001　ﾀｰﾄｩﾘｰｿｰﾎﾟﾝ　ｴｯｺｰﾝ

1 Introduction

Agriculture, fundamental for human sustenance and global food supply, confronts challenges like labor intensiveness and weather unpredictability, with variations such as droughts and floods threatening crop viability. Technology-driven transformations in modern agriculture, incorporating genetics, automation, and precision farming, bolster efficiency, diminish manual labor dependence[1] , and facilitate large-scale cultivation. Nevertheless, farming complexity persists due to natural variability, encompassing factors like crop choice, soil quality, irrigation, sunlight, and CO2 levels, significantly influencing outcomes and being susceptible to weather fluctuations. Urbanization's allure diminishes the rural workforce, resulting in fallow lands and diminished output, impacting global food security[2].

Amid the global push for automated harvesting systems, which leverage machine vision technology for precise fruit and crop harvesting, challenges persist in effectively synchronizing these robots with optimal environmental conditions. The vision systems of harvesting robots underscore the critical role of sunlight in green pepper harvesting, as it is essential for accurately identifying peppers amidst foliage. The preferred harvesting period typically spans from 8:00 AM to 4:30 PM[3], although this timeframe varies based on weather conditions and seasonality. Cloud cover, rainfall, and shortened winter daylight can restrict harvesting opportunities. However, relying solely on one type of camera poses limitations, particularly in adverse weather conditions. For instance, using an RGB camera is ineffective on rainy or cloudy days due to its reliance on light.

Conversely, relying solely on one camera can be challenging when the weather condition is insufficient for effective data collection. To reduce these challenges, researchers simultaneously employ both types of cameras to collect data. In this study, the researchers analyze the results based on RGB and IR images. Artificial Intelligence (AI) systems, integral to contemporary society, emulate and enhance human capacities through continuous learning. AI's utility, particularly in visually discerning tasks, is unparalleled. While human vision excels in object discrimination, AI's tireless, accurate operation is unmatched, leveraging training on labeled datasets through machine vision techniques like deep learning[4]. Additionally, the researchers utilize a Mask R-Convolutional Neural Network (CNN)[5] and Structural Similarity Index Measure (SSIM)[6] to enhance recognition and segmentation rates.

2 Materials and Methods

2.1 System Setup

Using two distinct cameras for data collection, the Intel Realsense D455 and the Optris XI400 presents unique challenges due to their differing dimensions and lens positions. Specifically, the Intel camera exhibits a deviation from the center, as shown in Fig 1, while the lens of the Optris camera aligns centrally within the device. Consequently, identifying the optimal photographic location becomes imperative, particularly considering the simultaneous capture of images from both sides of the green chili during data collection. Therefore, meticulous planning is required to ensure that the presence of one camera does not interfere with the data collection process of the other.

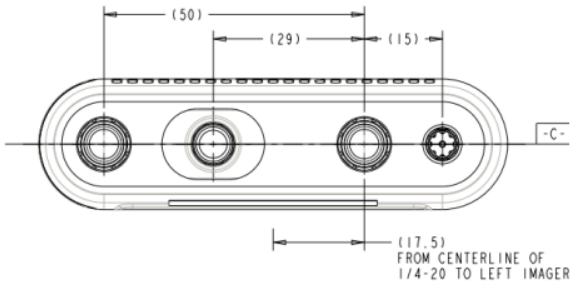

Fig. 1　Intel Realsense D455 lens position

2.2 Visual Sensing

Utilizing multiple cameras to capture identical scenes introduces challenges as the resulting images exhibit distinct characteristics due to variations in camera positions. Despite visual similarities, the inherent spatial differences hinder the images from identical, mainly when utilized for diverse processing needs. This discrepancy poses a significant issue as it can lead to calculation inaccuracies. Camera calibration becomes imperative, enhancing the precision of location detection and minimizing errors. The chosen approach involves applying the principles of triangulation within a stereo-vision system, as shown in Fig 2[7]. This method ensures a more accurate alignment of captured images, facilitating subsequent processing tasks. By adhering to the triangulation principle, the system aims to harmonize the spatial information from multiple cameras, enabling more reliable and consistent results in various applications and addressing the nuanced complexities of utilizing multiple cameras for image capture and subsequent analysis.

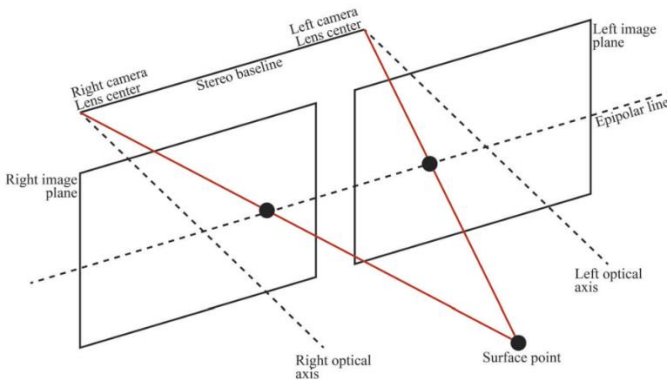

Fig. 2 Stereo vision system triangulation principle

The procedure above delineates a methodology tailored for parallel cameras within the same plane, sharing identical camera types. However, the forthcoming experiment deviates from this configuration, as it involves cameras set parallel to each other but positioned at an inclined angle, as shown in figure 3.

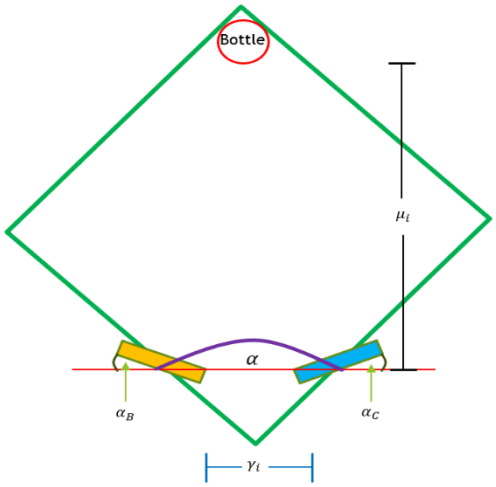

Fig. 3 RGB Angle and Distance test setup

Additionally, the cameras employed in this setup are of dissimilar types. Consequently, an additional equation is necessitated. Specifically, this Equation pertains to the angles of the two cameras oriented toward the object, denoted as Equation 1

$$\alpha = 180° - (\alpha_B + \alpha_C) \quad (1)$$

Moreover, there will be two additional variables γ_i which is the distance between the two types of cameras, and μ_i the distance between the camera and the object that collects data.

2.3 Image Acquisition

This investigation gathered a dataset comprising greenhouse green pepper images generously provided by KUT. The dataset incorporated diverse environmental conditions, encompassing sunny and cloudy days, and captured the subjects from various perspectives. The dataset employed in this study comprised a total of 4320 images, encompassing four distinct green pepper types and two different image modalities (RGB and IR) , as shown in Fig 4. The researcher judiciously distributed each category into training and validation sets to ensure a comprehensive evaluation, employing a randomized allocation method. Specifically, the training sets were designated for utilization during the model training phase, serving as the original input images for the training process. In contrast, the validation sets were reserved for assessing the model's performance after training. This meticulous dataset division into training and validation subsets facilitates a robust evaluation of the developed model under varying conditions, ensuring its efficacy and generalizability beyond the training data. provides a succinct representation of the randomized allocation of images across the training and validation sets for each category.

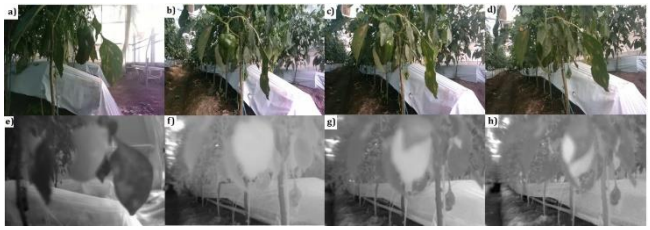

Fig. 4 Dataset of green pepper a) RGB left side covered by foliage 0%, b) RGB : right side covered by foliage 0%, c) RGB : right side covered by foliage 10-30%, d) RGB : right side covered by foliage >30%, e) IR : left side covered by foliage 0%, f) IR : right side covered by foliage 0%, g) IR : right side covered by foliage 10-30%, h) IR : right side covered by foliage >30%.

2.4 Image Preprocessing

Data augmentation was embraced to expand the sample size further to enhance the dataset's comprehensiveness, augment the feature information across various levels within the images, and improve the algorithm's adaptability to real-world scenarios. Specifically, the data augmentation technique employed in this investigation incorporated Laplacian sharpening. The utilization of Laplacian sharpening serves to enhance image sharpness, rendering object edge details within the image more distinct. Additionally, it addresses issues arising from unclear images due to low resolution. The application of the Laplace operator, integral to Laplacian sharpening, is instrumental in achieving these improvements. Laplacian sharpening encompasses the application of the Laplacian operator to an image[8]. The Laplacian operator, symbolized as ∇^2, constitutes a second-order derivative and is frequently expressed in mathematical terms as follows equation 2

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \quad (2)$$

Within the domain of image processing, the technique of Laplacian sharpening entails the deduction of the outcome derived from applying the Laplacian operator to the original image from the original image. This process generates a sharpened image, denoted as g and can be formally articulated as follows equation 3

$$g(x,y) = f(x,y) - \nabla^2 f(x,y) \quad (3)$$

$g(x,y)$ is the sharpened image.
$f(x,y)$ is the original image.

In this context, where x and y represent pixel coordinate values, $g(x,y)$ signifies the resulting sharpened image, $f(x,y)$ denotes the original image, $\nabla^2 f(x,y)$ represents the Laplace transform of the original image, and the Laplace mask is visually presented in equation 4 and results after image sharpening, as shown in Fig 3

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (4)$$

Fig. 5 Image sharpening a) Original RGB image, b) RGB after sharpening, c) Original IR image,) IR after sharpening

2.5 Data Annotation

The image annotation process holds pivotal significance in training models, as it involves delineating object boundaries to ensure the specificity of model training towards desired objectives. In this context, the annotation tool employed is Labelme. The experimental dataset was annotated using Labelme to produce mask images corresponding to the delineation of green peppers within the images. Furthermore, evaluating the trained model's performance in instance segmentation involved a comparative analysis between the annotated mask images and the model's predicted mask outputs. Specifically, regionsas of the images corresponding to green peppers were meticulously labeled, while the remaining areas were designated as background. The resultant annotated images depict the labeled regions of green peppers, as shown in Fig 4.

Fig. 6 RGB and IR image with label and mask box

2.6 Mask R-CNN

Mask R-CNN, an abbreviation for Mask Region-based Convolutional Neural Network, stands at the forefront of contemporary computer vision research, exhibiting remarkable prowess in instance segmentation. Introduced as an extension of the Faster R-CNN architecture, Mask R-CNN seamlessly integrates object detection and segmentation, enabling precise delineation of object boundaries and identifying distinct instances within an image[9]. The fundamental innovation within Mask R-CNN lies in its ability to concurrently generate pixel-level masks for each object instance while performing object detection. This task amalgamation is accomplished by incorporating a dedicated mask branch parallel to the existing branches for object classification and bounding box regression. Leveraging a two-stage approach, Mask R-CNN initially proposes region proposals through the Region Proposal Network (RPN) and subsequently refines these proposals with refined bounding box coordinates and corresponding instance masks. The architecture's robustness is underscored by its capacity to handle various object scales and shapes, rendering it highly adaptable to complex scenes. Mask R-CNN has proven instrumental in various applications, ranging from medical image analysis to autonomous vehicles, owing to its proficiency in extracting fine-grained spatial information. As a testament to its efficacy, Mask R-CNN has emerged as a cornerstone in instance segmentation, embodying the paradigm shift towards comprehensive visual scene understanding and semantic segmentation in academic ,industrial spheres and an overview of Mask R-CNN, as shown in Fig 7.

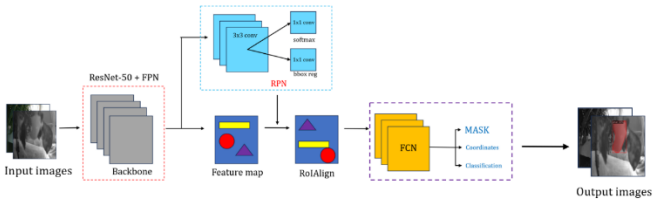

Fig. 7 Overview of Mask R-CNN

2.7 Feature Extraction and ROI

The establishment of deep neural network models with varied depths is accomplished by designing different weight layers. AlexNet, ZF, VGG, GoogleNet, and ResNet are prominent models in deep neural networks[10]. Although deeper networks have the potential to yield higher accuracy, a trade-off exists with a reduction in model training and detection speeds. ResNet, notable for its residual structure mitigating challenges such as gradient disappearance and training degradation without increasing model parameters, has been chosen as the foundational network for feature extraction in this research. This research employs two types of cameras, resulting in two distinct pixel configurations for RGB and IR images of a single green pepper. FPN outputs for different levels are designed to accommodate these varying image scales. The study focuses on analyzing single green peppers using two distinct imaging modalities: RGB and Infrared (IR). The base image size is standardized for the RGB images at 720x1280 pixels. Feature Pyramid Network (FPN) outputs at different levels are meticulously tailored to accommodate these specifications:

FPN output level 2-5 are 1/4, 1/8, 1/16 and 1/32 and dimensions size are 180x320, 90x160, 45x80, 22.5x40 pixels accordingly. Concurrently, Infrared (IR) images of single green peppers are acquired with a base image size of 288x382 pixels. The FPN outputs at different levels for IR images are adjusted accordingly: 1/4, 1/8, 1/16, 1/32, and the dimensions' sizes are 72x96, 36x48, 18x24, and 9x12 pixels. These meticulously designed imaging scales and FPN outputs are integral to generating Region of Interest (RoI) for subsequent analysis and facilitate the effective representation and detection of features in the study's context of single green peppers. In RoI generation, the aspect ratio of labeled rectangular boxes for single and occluded green peppers is approximately 1:1, determined by bounding box definition using minimum and maximum coordinates in both x and y directions. The FPN outputs play a crucial role in this process, offering tailored information for generating RoIs.

2.8 Image Segmentation and Loss Function

The RoIAlign-generated feature maps were subsequently processed through fully convolutional network. The utilization of fully convolutional network was threefold, encompassing classification, bounding box regression and coordination. Conversely, applying fully convolutional network was dedicated to segmenting individual instances of green peppers. The classification task involved feeding the outputs from the fully connected network into a Softmax layer, thereby obtaining the classification probabilities. Simultaneously, the convolutional layers were employed for the intricate instance segmentation process. The training of the network entailed the establishment of a loss function, which quantified the disparities in the network prediction. Assuming P_{class} , P_{bbox} , and P_{mask} represent the predicted class probabilities, bounding box coordinates, and mask predictions, respectively, and T_{class} , T_{bbox} , and T_{mask} represent the corresponding values, the loss function can be written as equations 5,6 and 7

Classification Loss (Cross-Entropy):

$$L_{class} = -\frac{1}{N_{roi}}\sum_{i=1}^{N_{roi}}\sum_{c=1}^{C} T_{class,i,c}\log\left(P_{class,i,c}\right) \quad (5)$$

Bounding Box Regression Loss (Smooth L1):

$$L_{bbox} = \frac{1}{N_{roi}}\sum_{i=1}^{N_{roi}}\sum_{j\in\{x,y,w,h\}} smooth_{L1}(P_{bbox,i,j} - T_{bbox,i,j}) \quad (6)$$

Mask Segmentation Loss (Binary Cross-Entropy):

$$L_{mask} = -\frac{1}{N_{roi}}\sum_{i=1}^{N_{roi}}\sum_{p=1}^{(H)(W)}[T_{mask,i,p}\log\left(\sigma\left(P_{mask,i,p}\right)\right) + (1 - T_{mask,i,p})\log\left(1-\sigma(P_{mask,i,p})\right)] \quad (7)$$

N_{roi}denotes the count of regions of interest (RoI), C signifies the number of distinct classes, and Hand W corresponds to the height and width of the precited mask, respectively. Additionally, σ represents the sigmoid function. Comprehensive loss is formulated as the cumulative sum of individual losses, incorporating potential weighting coefficients is presented in equation 8.

Total Loss:

$$L_{total} = \lambda_{class}L_{class} + \lambda_{bbox}L_{bbox} + \lambda_{mask}L_{mask} \quad (8)$$

It is noteworthy that the user has the flexibility to fine-tune the weighting coefficients (λ_{class}, λ_{bbox}, λ_{mask}) based on the relative significance of each constituent in the specific context of their application. This flexibility allows for the customization of the loss function to best align with the task's prioritized aspects.

2.9 Edge Detection

Edge detection is a fundamental technique in computer vision and image processing that aims to identify boundaries and transitions within images, highlighting regions were intensity or color changes sharply. These boundaries represent the contours or edges between distinct objects or structures in the visual content. The primary objective of edge detection is to enhance the visibility of these essential features, enabling subsequent analysis, segmentation, and recognition tasks in computer vision applications. Various algorithms are employed for edge detection, each with their approach and characteristics. Popular methods include the Sobel and Prewitt operators, which emphasize gradient changes in horizontal and vertical directions, and the Canny edge detector, known for its multi-stage process that minimizes false positives. Other techniques, such as the Laplacian of Gaussian (LoG) and Kirsch operator, leverage convolution and mathematical operations to identify edges based on image intensity variations. Edge detection is a critical step in image processing pipelines, serving as a foundation for tasks like object recognition, image segmentation, and feature extraction. Its application is widespread in fields such as medical imaging, autonomous vehicles, surveillance, and industrial quality control, where accurate delineation of objects and structures within images is essential for robust and precise computer vision analyses.

2.10 Edge Detection Algorithm

Edge detection algorithms are essential components in computer vision, focusing on identifying abrupt variations in image intensity. The Canny edge detector, a prominent method, employs gradient calculation through convolution filters, emphasizing both horizontal and vertical changes. Subsequent non-maximum suppression isolates local maxima, and hysteresis-based edge tracking discerns solid and weak edges, producing a binary edge map delineating structural boundaries. Other methodologies, including Sobel and Prewitt operators, utilize gradient information to accentuate directional changes. These algorithms play a crucial role in image processing tasks, such as object recognition and segmentation, where accurate demarcation of object boundaries is imperative. By enhancing the visibility of significant transitions in visual data, edge detection algorithms contribute significantly to feature extraction and pattern recognition within diverse computer vision applications. The following enumeration delineates noteworthy edge-detection algorithms

Roberts – Roberts Cross edge detection is a simple and computationally efficient method for detecting edges in images. It involves convolving the image with a pair of 2x2 convolution kernels[11]. The Roberts Cross operator equations can be written as equations 9 and 10

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} * I \quad (9)$$

$$G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} * I \quad (10)$$

Where: I is the input image , * denotes the convolutional operation.
The resulting gradient images, Gx and Gy , capture intensity changes in the horizontal and vertical directions, respectively. The gradient magnitude G at each pixel is calculated using the formula:

$$G = \sqrt{Gx^2} + Gy^2 \quad (11)$$

Sobel – Sobel edge detection is a widely used method in computer vision for highlighting edges in an image by emphasizing changes in intensity in both the horizontal and vertical directions[12]. The Sobel operator involves convolving the image with 3x3 kernels, one for detecting changes in intensity in the horizontal direction Gx and the other for the vertical direction G_y. The resulting gradient images, Gx and G_y, are combined to obtain each pixel's gradient magnitude G and direction θ, The Sobel operator can be written as equations 12, 13, 14 and 15

$$Gx = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I \quad (12)$$

$$Gy = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I \quad (13)$$

$$G = \sqrt{Gx^2} + Gy^2 \quad (14)$$

and the gradient direction θ is given by:

$$\theta = \arctan\left(\frac{G_y}{G_x}\right) \quad (15)$$

The Sobel edge detection algorithm effectively highlights edges by accentuating intensity changes in the image along both the horizontal and vertical axes.

Prewitt – Prewitt edge detection is a method commonly employed in image processing for highlighting edges by emphasizing changes in intensity along both horizontal and vertical directions. Proposed by Judith M. S. Prewitt[13], this technique employs convolution with Prewitt kernels to calculate image gradients, offering a simplified alternative to more complex operators. The Prewitt operator equations for horizontal Gx and Gy gradients are represented by equations 16 and 17

$$Gx = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} * I \quad (16)$$

$$Gy = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} * I \quad (17)$$

here, I represents the input image, and $*$ denotes the convolution operation. The gradient magnitude G at each pixel is computed as $G = \sqrt{Gx^2} + Gy^2$, providing a measure of intensity changes in the image. Prewitt edge detection proves valuable for its simplicity and efficiency in capturing edge information along multiple directions, contributing to applications such as image segmentation and feature extraction in computer vision tasks.

Laplacian – The Laplace operator ∇^2 is a mathematical operator commonly utilized for edge detection in image processing. It is applied through convolution with a Laplacian kernel, represented by specific convolution matrices[14]. The 3x3 Laplacian kernel can be written as equations 18 and 19

$$\nabla^2 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} * I \quad (18)$$

Here, the central element (-4) represents the weight assigned to the pixel being processed, while the neighboring elements (1) indicate the weights of surrounding pixels. For the 5x5 Laplacian kernel, the formulation is:

$$\nabla^2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} * I \quad (19)$$

In this case, the central element (-16) represents the weight assigned to the pixel being processed, while the surrounding elements (1 or 2) denote the weights of neighboring pixels. The convolution operation * is applied to the image I, producing the Laplacian response. This response emphasizes regions where intensity changes abruptly, facilitating effective edge detection in image analysis and processing tasks.

Canny Edge detection – Canny edge detection is a sophisticated image processing technique designed to identify and highlight edges in an image, minimizing the influence of noise. Proposed by J. Canny in 1986[15], this method involves multiple stages to achieve robust edge detection. The key steps include:

Gradient Calculation: Compute the image gradient using convolution with Sobel filters to emphasize changes in intensity in the horizontal and vertical directions.
Gradient Magnitude and Orientation: Determine the gradient magnitude and orientation at each pixel.
Non-Maximum Suppression: Suppress non-maximum gradient values to retain only local maxima along the edges.
Edge Tracking by Hysteresis: Establish high and low thresholds for gradient magnitudes, identify pixels with gradient magnitudes above the high threshold as strong edge points, connect weak edge points to strong edge points if they are part of the same edge structure.
Mathematically, the gradient magnitude G is calculated as $G = \sqrt{Gx^2} + Gy^2$, where Gx and Gy are the horizontal and vertical gradients. The gradient orientation θ is determined as $\theta = \arctan\left(\frac{G_y}{G_x}\right)$. The Canny edge detection equation incorporates these principles, providing an effective approach for accurate edge localization and noise reduction in various computer vision applications.

2.11 Structural Similarity (SSIM)

In the comparative analysis of two images facilitated by a software system, the Structural Similarity Index (SSIM) principles are applied to ensure a comprehensive evaluation that transcends mere correlation coefficients. The first step involves mitigating the impact of brightness on structural information. Luminance information is subtracted during the calculation of structural information, and subsequently, the mean value of the image is subtracted. This initial adjustment aims to preserve the inherent structural characteristics of the fruits depicted in the images. Subsequently, the structural information is further refined to eliminate the influence of image contrast. Normalization of the variance of the images is undertaken during the computation of structural details. This step ensures that the structural features are assessed independently of variations in image contrast, contributing to a more precise analysis. The final phase involves the comprehensive calculation of structural information, incorporating the outcomes of brightness and contrast comparisons[16]. The conventional approach of calculating correlation coefficients is augmented to account for the nuanced impact of brightness and contrast on image dissimilarity. The overarching workflow of this SSIM process is delineated in Fig 7, emphasizing the sequential application of these principles in achieving a holistic evaluation of image similarity. By systematically addressing the influence of luminance and contrast, the SSIM methodology ensures a refined and nuanced assessment, offering a more accurate depiction of the similarity between two images within the software system.

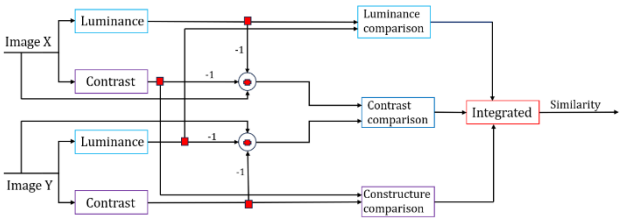

Fig. 8 SSIM structure workflow

The Structural Similarity Index (SSIM) comprises three constituent sub-indices: the luminance index, contrast index, and structure index. Luminance, within the context of the SSIM index, pertains to the intensity of the object portrayed in the image, delineated by pixel values. The luminance index, therefore, serves as a metric for capturing the inherent brightness characteristics of the recorded object within the image. The contrast index encapsulates the discernible difference in luminance or the extent of luminance variation across the image. This index provides a quantitative measure of the variability in luminance values, offering insights into the image's overall contrast properties. As a component of the SSIM, the structure index reflects the Pearson correlation of luminance between two images, namely, image X and image Y. This index evaluates the similarity in the structural patterns of luminance across corresponding points in the images. The comparison functions for luminance, contrast, and structure at each point in the images are expressed through equations 20, 21, and 22, respectively. These equations encapsulate the mathematical formulations employed to quantify the luminance intensity, contrast variation, and structural correlation, forming the basis for a comprehensive evaluation of image similarity within the SSIM framework.

Luminance: $l(x,y) = \frac{2\mu_x\mu_y+C_1}{\mu_x^2+\mu_y^2+C_1}$ (20)

Contrast: $c(x,y) = \frac{2\sigma_x\sigma_y+C_2}{\sigma_x^2+\sigma_y^2+C_2}$ (21)

Structure: $s(x,y) = \frac{\sigma_{xy}+C_3}{\sigma_x\sigma_y+C_3}$ (22)

In the Structural Similarity Index (SSIM) formulation, μ_x and μ_y represent the local means, σ_x and σ_y denote the standard deviations, and σ_{xy} signifies the cross-covariance between image X and image Y. The equations 23, 24, and 25 express the mathematical definitions of μ_x, σ_x, and σ_{xy}. To ensure computational stability and prevent division by minute denominators, C_1, C_2, and C_3 function as regularization constants with diminutive values. The introduction of these constants is imperative for mitigating potential numerical instabilities in SSIM calculations, thereby reinforcing precision and robustness in diverse image-processing contexts.

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (22)$$

$$\sigma_x = \left(\left(\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_x)^2\right)^{\frac{1}{2}}\right. \quad (23)$$

$$\sigma_{xy} = \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_x)(y_i-\mu_y) \quad (24)$$

The index i represents all points within a localized region. At the same time, N signifies the total number of points encompassed by this area, including the evaluating point and its N neighboring points. The configuration of the local area is adaptable, allowing for adjustments in shape and size through the selection of filter types, such as the Gaussian filter and filter size. Ultimately, the Structural Similarity Index (SSIM) integrates three sub-functions, culminating in its final formulation as shown in equations 25 and 26. These equations encapsulate the SSIM index's mathematical representation, a composite measure derived from the interplay of these sub-functions.

$$SSIM(x,y) = [l(x,y)]^{\alpha}[c(x,y)]^{\beta}[s(x,y)]^{\gamma} \quad (25)$$

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+C_1)(2\sigma_{xy}+C_2)}{(\mu_x^2+\mu_y^2+C_1)(\sigma_x^2+\sigma_y^2+C_2)} \quad (26)$$

An SSIM index attaining 1 signifies perfect concordance between two images, whereas an SSIM index below 1 indicates a disparity between the compared images. The overall SSIM and its sub-indices for the compared images are computed as the mean values of their respective index maps. This analytical approach allows for a quantitative assessment of image similarity, with 1 indicating a complete match and values lower than one indicating deviations or dissimilarities between the compared images. Using mean values in the computation contributes to a comprehensive and representative evaluation of the images' structural, luminance, and contrast attributes.

2.12 Validation and Analysis

The F1 score, a pivotal metric in image processing, is prominently employed in binary classification scenarios such as object detection, image segmentation, and classification tasks. It amalgamates the principles of precision and recall, offering a comprehensive assessment of a model's efficacy in delineating and identifying objects within images. In image processing, particularly in tasks involving segmentation or object detection, precision denotes the accuracy with which the model correctly identifies relevant regions. At the same time, recall measures the model's ability to encompass all pertinent instances. Precision is the ratio of accurate optimistic predictions to the sum of true and false positives. At the same time, recall is expressed as the ratio of true positives to the sum of true positives and false negatives. The F1 score, the harmonic mean of precision and recall, is a harmonized metric that encapsulates the impact of false positives and false negatives. Its formula delineates a balanced evaluation considering both precision and recall. Symbolically, the F1 score equations are represented as equation 27 to 29

$$P = \frac{True\ Positives}{True\ Positives + False\ Positives} \quad (27)$$

$$R = \frac{True\ Positives}{True\ Positives + False\ Negatives} \quad (28)$$

$$F1 = 2\frac{Ture\ Positives}{True\ Positives + False\ Negatives} \quad (29)$$

3 Experimental

3.1 Camera Experimental

The experimental procedures are organized into phases:

- Evaluation of the angular orientation of RGB camera during this phase, an isolated evaluation of the RGB camera will be conducted , determining the threshold at which data collection from the object commences. In the experimental evaluation, there is a necessitated adjustment in camera orientation due to the off-centered placement of the RGB camera lens within the camera structure and the angular orientation of each camera is systematically varied within the range of 0 to 90 degrees. Alterations are implemented at increments of 5 degrees, guided by Equation 1, wherein the values $\alpha_B$and α_C will consistently share the same degree values throughout the testing process.
- Assessment of the maximum distance achievable by each camera for data collection γ_i, ensuring that the camera does not capture images of other cameras within its field of view. The experimental assessment encompasses distances ranging from 15 to 30 centimeters.
- The conclusive phase of the experiment involves utilizing the values of α and γ_i to determine the optimal shooting distance, denoted as μ_i . This optimization considers both the shooting distance and the angles of both camera types. During image capture, it is imperative to ensure no overlap between the field of view of one camera type and another, maintaining distinct visibility for each camera type.

3.2 Mask R-CNN + SSIM

The subsequent phase of the study involves practical assessments utilizing an authentic camera within a greenhouse environment to acquire empirical data for further investigation. Data collection transpired on the dates 11/29, 11/30, and 12/1. The initial step encompassed the systematic recording of data as Fig 4. This involved capturing images in four RGB types (a-d) and four IR types (e-h), each associated with specific locations as shown in Fig 9 and green pepper 540 images were recorded for each type, resulting in 4320 images. Furthermore, each image type was subdivided into three sets: 1) training set, 2) validating set, and 3) testing set. The experimental procedures are methodically organized into distinct phases.

- The initial phase of the study involved data annotation through the utilization of labelme. Subsequently, the annotated data was employed to assess the accuracy of the Mask R-CNN system algorithm. This iterative process facilitated evaluating the algorithm's performance in processing annotated data, providing insights into its efficacy and precision in handling the specific task. The systematic data annotation through labelme was a foundational step in preparing the dataset for algorithmic testing and performance evaluation.
- In the second procedural step, images obtained from the segmentation process in step 1 were utilized for testing purposes. Specifically, the acquired data were employed in two distinct scenarios: first, employing infrared (IR) images as the testing dataset for models trained with red-green-blue (RGB) images, and second, using RGB images as the testing dataset for models trained with infrared (IR) images. This methodology aimed to assess the efficacy of the segmentation algorithm in identifying green peppers under varying conditions, specifically evaluating its ability to generalize across different imaging modalities and ascertain the robustness of the model in pepper identification.
- In the third procedural step, after obtaining images from the initial segmentation step, an analytical procedure was employed utilizing the Structural Similarity Index (SSIM) method. This method entailed a comparative analysis akin to the approach in step 2, involving the juxtaposition of infrared (IR) images with red-green-blue (RGB) images and vice versa. The objective was to ascertain the discernibility of green peppers through the SSIM method, thereby evaluating the effectiveness of the segmentation process in distinguishing these peppers based on variations in imaging modalities. This analytical step contributed to the comprehensive assessment of the algorithm's performance in identifying green peppers across different image representations.
- In the fourth procedural step, this process resembles step 3, albeit explicitly focusing on a designated point of interest. Unlike the comprehensive image analysis in the prior step, the comparison is selectively confined to the region delineated by the bounding box. This targeted approach aims to streamline the computational workload by restricting the assessment to solely those images within the specified area. By doing so, the intent is to mitigate data volume, enhance computational efficiency, and diminish extraneous noise that might

arise from considering the entire image, thereby refining the precision of the evaluation.

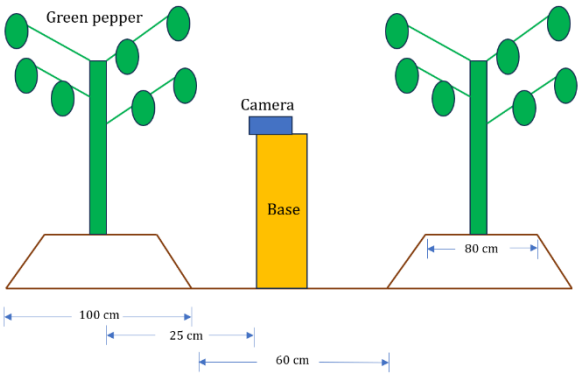

Fig. 9 Data collection information setup in greenhouse

3.3 Mask R-CNN + Edge Detection + SSIM

Initial Method

- In the initial phase, the procedure involves the demarcation of object boundaries and the application of a masking technique to isolate the object box. The forthcoming experimentation will build upon the data acquired through the preceding methodology, which incorporates Mask R-CNN and SSIM. Specifically, we intend to augment this approach by integrating an Edge Detection step before the SSIM process, as delineated in Figure 10.The Edge Detection procedure encompasses five distinct methods, as elucidated in the antecedent chapter. These methods comprise Robert, Sobel, Prewitt, Laplacian, and Canny Edge Detection. Subsequently, the highest values obtained from the three most effective methods will be utilized in the ensuing process.

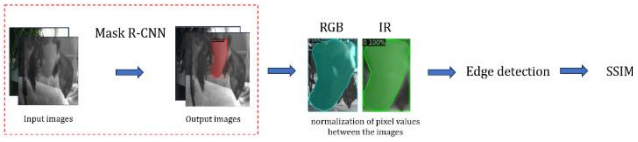

Fig. 10 Initial step method overview

Secondary Method

- In the second step, these top-performing methods from the first step will be reapplied in the Edge Detection process after initial noise reduction. This phase involves a nuanced comparison with the Mask R-CNN step. Unlike the conventional approach in Mask R-CNN, which consists in masking objects and applying label painting, the modified procedure in this step employs the Mask and label without painting over the object. The image size is adjusted uniformly, and a crop operation is executed to retain only the desired objects, as shown in Fig 11. Subsequently, the Edge Detection process is initiated, followed by the SSIM evaluation, as shown in Fig 12. This systematic approach aims to capitalize on the superior performance of selected Edge Detection techniques while incorporating insights from the initial noise reduction steps, thereby refining the overall image processing methodology.

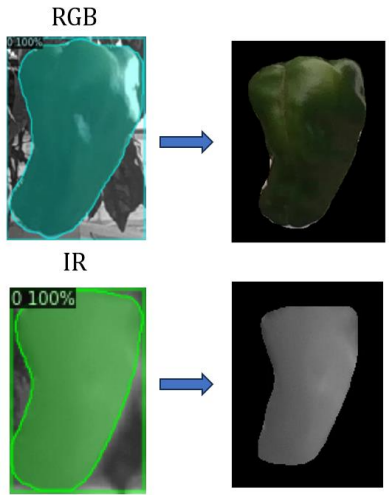

Fig. 11 Crop Objection in bounding box

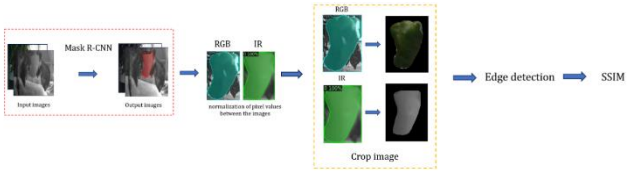

Fig. 12 Secondary step method overview

4 Results

4.1 Camera Results

- The experimental findings indicate that photographing at various angles before and after the side-switching operation proved unproblematic within the range of 0 to 75 degrees. However, deviations arose when the angle exceeded 80 degrees, leading to the object's image extending beyond the frame boundaries.
- Concerning the range of cameras capable of initiating object capture image, this capability extends from γ_i = 0 to 30 cm within the camera's plane at 0 degrees. This ensures unimpeded photographing without including another camera type within the frame of the image.
- The first two experiments were amalgamated and tested collectively in the concluding phase. The outcomes indicate that for the left side, with the RGB camera on the left and the IR camera on the right, images can be captured without interference from other camera types within the range of γ_i = 15.5 - 22 cm. The parameters $\alpha = 120°$, α_B and α_C are set within 0-30 degrees. Similarly, for the right side, with the RGB camera on the right and the IR camera on the left, images can be taken without incorporating other camera types within the range of γ_i = 15.5 - 22 cm, with the same angular constraints.
- Regarding μ_i, the RGB camera exhibits no issues and can capture images within the 15-30 cm range. However, the IR camera encounters challenges when the camera-object distance exceeds 28 cm, manifesting barrel distortion symptoms. The subsequent summary section will provide further elucidation on barrel distortion symptoms.

4.2 Mask R-CNN + SSIM Results

- In the initial phase, employing the Labelme annotation method and testing with the Mask R-CNN algorithm on red-green-blue (RGB) images yielded an annotation accuracy of 0.976. In contrast, a corresponding accuracy of 0.989 was achieved when testing on infrared (IR) images. This quantitative assessment reflects the algorithm's proficiency in accurately identifying and annotating regions of interest within RGB and IR images, with higher values indicating greater precision in the annotation process. The numeric outcomes provide quantitative insights into the algorithm's performance during the image annotation stage, contributing to the overall evaluation of its efficacy under different imaging conditions.
- During the second step, employing infrared (IR) images as testing data for red-green-blue (RGB) models, and vice versa, yielded outcomes with a discernible absence of accuracy, registering a score of 0. This denotes a need for more precision in the models' ability to correctly classify and identify objects when confronted with testing data from an alternate imaging modality. The null accuracy values underscore the challenges and limitations encountered when attempting cross-modal testing, signifying the necessity for further refinement and adaptation of the models to enhance their capacity for generalized object recognition across different spectral domains.
- In the third step, the introduction of the Structural Similarity Index (SSIM) into the image comparison methodology resulted in SSIM scores ranging approximately between 0.20 and 0.25 when comparing infrared (IR) and red-green-blue (RGB) images. This quantitative assessment reflects the degree of structural similarity between the two modalities, with the SSIM scores measuring the likeness in structural patterns. The observed scores in this range suggest a moderate level of similarity, indicating that the structural characteristics of the IR and RGB images exhibit discernible differences while still possessing certain standard features, as quantified by the SSIM.
- In the fourth step, following a methodology analogous to the third step, the emphasis was explicitly directed toward a designated object within a bounding box. The application of the Structural Similarity Index (SSIM) to compare infrared (IR) and red-green-blue (RGB) images, limited to this defined region, yielded SSIM scores ranging approximately between 0.4 and 0.45. This targeted comparison within the bounding box indicates a moderate increase in the SSIM scores compared to the comprehensive image assessment in the third step, suggesting a higher level of structural similarity when focusing solely on the specified object within the bounding box.

4.3 Mask R-CNN + Edge Detection + SSIM Results

- Upon completion of the initial testing phase, the obtained results are as follows: Canny Edge Detection, Roberts, Laplacian 3x3, Sobel, and Laplacian 5x5. The corresponding Structural Similarity Index (SSIM) scores, arranged in descending order, are 0.577, 0.552, 0.551, 0.44, and 0.269. These results are visually presented in Fig 13, showcasing the highest scores. Notably, the application of Edge Detection in this method yielded an SSIM score of 0.577

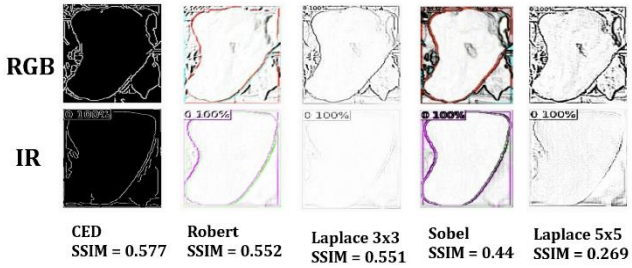

Fig. 13 SSIM score comparison with Edge detection methods

- Based on the outcomes of the initial experiment, three methods emerged with the highest Structural Similarity Index (SSIM) scores: Canny Edge Detection, Roberts, and Laplacian 3x3, scoring 0.577, 0.552, and 0.551, respectively. In the second step, these top-performing methods from the first step will be reapplied in the Edge Detection process after initial noise reduction. This phase involves a nuanced comparison with the Mask R-CNN step. Unlike the conventional approach in Mask R-CNN, which consists in masking objects and applying label painting, the modified procedure in this step employs the Mask and label without painting over the object and the outcomes derived from the second experiment assess the performance metrics associated with the recently introduced Mask R-CNN. Within the RGB and IR cropping segments, the acquired F1 scores stand at 0.7894 and 0.9815, respectively. Furthermore, applying the new method for edge detection results in images, as exemplified in Fig 14. In contrast, the SSIM segment yields analogous scores of 0.7894 and 0.9815. Notably, the recorded scores predominantly concentrate within the range of 0.790 to 0.810., thereby encapsulating a substantial portion of the experimental outcomes, as shown in Fig 15.

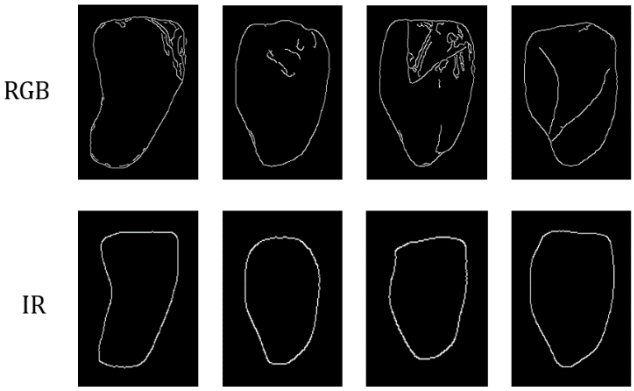

Fig. 14 RGB and IR with Canny Edge Detection , from left to right are capture from the left side, capture from the right side without a foliage, capture from the right with a foliage of 10-30%, capture from the right with a foliage more than 30%.

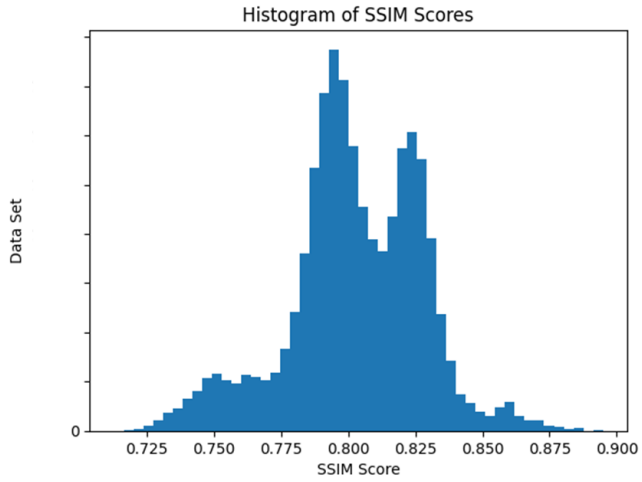

Fig. 15 Secondary step method SSIM score

5 Summary

5.1 Camera

The experimental findings suggest that the optimal shooting angle for capturing images without other cameras in the frame is 30 degrees. Maintaining a distance between cameras ranging from 15.5 to 22 cm is recommended, with a preference for the lower limit of 15.5 cm for space efficiency in potential installations on an automated harvesting robot. The permissible range for camera-object distance extends from 15 to 30 cm without encountering issues in the RGB camera. However, in the case of IR cameras, complications arise beyond 28 cm, leading to a phenomenon known as barrel distortion Fig 16 , akin to a fisheye lens effect, as shown in Fig 17[17]. Commonly associated with wide-angle lenses, this distortion can be rectified through algorithms, such as the Correction of Barrel Distortion in Fisheye Lens Images Using Image-Based Estimation of Distortion Parameters by M. Lee[18] or T. Hwan Kim's An Efficient Barrel Distortion Correction Processor for Bayer Pattern Images[19]. However, due to the significantly lower resolution of the IR camera 382x288 pixels[20] compared to the RGB camera 1280x800[21], image editing complexities arise in the IR domain. Consequently, employing the IR camera within a shooting distance of less than 28 cm is recommended for expeditious and straightforward resolution, thereby mitigating challenges and noise during subsequent image processing.

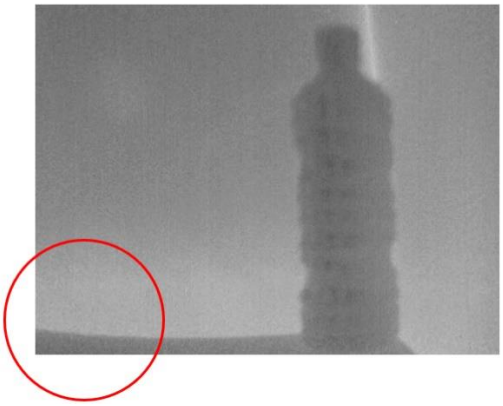

Fig. 16 Barrel distortion of IR image when range distance exceeds 28 cm

Fig. 17 Image of Brick wall captured with wide angle lens

5.2 Mask R-CNN + SSIM

Upon conducting testing, several observations emerged regarding the efficacy of the methodology. In the second method, the interchangeability of infrared (IR) and red-green-blue (RGB) images for testing proved unfeasible due to numerous constraints and image-specific conditions. However, in the third step, where the Structural Similarity Index (SSIM) was incorporated for image comparison, some degree of success was achieved despite the relatively low scores. Notably, focusing exclusively on the region of interest within a bounding box in the fourth step yielded favorable outcomes. The SSIM score exhibited a significant improvement, escalating from 0.25 to 0.45. This progression underscores the pivotal role of noise reduction in enhancing SSIM scores, emphasizing the importance of concentrating solely on the target object within the bounding box for optimal results, particularly when assessing structural similarities in images.

5.3 Mask R-CNN + Edge Detection + SSIM

Both experiments revealed that the optimal approach involves mitigating image noise before SSIM-based comparisons. Highly detailed RGB images exhibit a modest correctness probability of merely 0.7984. In contrast, IR images, characterized by reduced image intricacies, achieve a notably higher score of 0.9815, surpassing the RGB score of 0.1831. Despite the superior IR score, a notable issue arises during the edge detection phase: the images acquired from the IR camera struggle to detect leaves due to insufficient image details. This contrasts RGB images that boast discernible leaf lines, contributing to object coverage. Such intricacies may pose future challenges, particularly if objects become conglomerated and need clear demarcation.

6 Conclusions

This research is focused on advancing methodologies for effectively detecting green peppers, particularly within the controlled environment of a greenhouse. A comprehensive analysis of green peppers grown in the greenhouse environment has revealed various challenges that necessitate nuanced solutions. One of the primary challenges identified is the varied positioning of green pepper fruits throughout the plant. Given that green pepper plants can produce fruit across the entirety of the plant, the resulting fruits exhibit a range of heights, some elevated and others at lower levels.

This inherent variability in fruit positioning introduces complexities in subsequent detection processes. Moreover, the abundance of leaves on green pepper plants and their dense arrangement further complicates distinguishing individual green peppers from the foliage. The inherent tendency of green pepper fruits to grow nearby, forming clusters, poses an additional layer of difficulty in achieving accurate and precise detection. The challenge is exacerbated by the fact that individual fruits must be isolated during the data collection, preventing them from being clustered with other fruits. The research underscores the importance of meticulous data collection to address these challenges. The optimal approach involves maintaining 25cm from the object of interest and tilting the camera at a 30-degree angle. These parameters ensure the collected data is well-positioned, avoiding clustering issues and enabling accurate separation of individual green peppers from the surrounding foliage. However, even with careful data collection, challenges persist in accurately discerning the color of green peppers, leaves, and fruits, mainly when relying solely on a conventional RGB camera. The similarity in color poses difficulties in differentiation. As a potential solution, the research explores using an infrared (IR) camera, which exhibits promise in classification but encounters challenges related to accuracy, particularly in cases where the temperature of the fruits and leaves is similar.

The study employs the Mask R-CNN process to analyze and detect green peppers, achieving commendable accuracy with scores of 0.976 and 0.989 for different process aspects. However, the subsequent structural similarity index (SSIM) process presents distinct challenges. Despite the RGB image scoring marginally lower 0.1831 than the IR image, it encounters more intricate challenges. The high resolution of the RGB image facilitates the differentiation of fine details in the leaves, which may obscure the green peppers. Conversely, the images obtained from the IR camera struggle to distinguish leaves that may obscure the green pepper fruits. Practical challenges also extend to the physical setup within the greenhouse. Walkways composed of dirt necessitate frequent adjustments to the camera position to ensure a level and consistent perspective. The computational aspect of the process introduces another layer of complexity, with the calculation process involving numerous steps and substantial processing time. For instance, calculating a single result requires up to 16 hours, underscoring the need for more efficient computational methodologies for practical applications. In conclusion, the research highlights the multifaceted challenges of detecting green peppers in a greenhouse environment. Each aspect requires careful consideration and innovative solutions, from nuanced data collection to color differentiation and computational efficiency. Addressing these challenges is pivotal for practically implementing the methodology in real-world scenarios, were efficiency and accuracy.

References

(1) Erenstein, O., Chamberlin, J., & Sonder, K. (2021). “Farms Worldwide: 2020 and 2030 Outlook”, in Proceeding of the Outlook on Agriculture, Vol. 50(3), pp. 221-229.

(2) https://www.maff.go.jp/j/tokei/sihyo/data/08.html

(3) Nan, Y., Zhang, H., Zeng, Y., Zheng, J., & Ge, Y. (2022). Faster and Accurate Green Pepper Detection Using NSGA-II-based Pruned YOLOv5l in the Field Environment. Computers and Electronics in Agriculture, Dec 2022.

(4) Zhou, W., Cui, Y., Huang, H., Huang, H., Wang, C. (2024). A Fast and Data-Efficient Deep Learning Framework for Multi-class Fruit Blossom Detection. Computers and Electronics in Agriculture, Vol.217, Jan 2024.

(5) Ganesh, P., Volle, L., Burks, T. F., Mehta, S. S. (2019). Deep Orange: Mask R-CNN based Orange Detection and Segmentation. IFPA Conference Paper Archive, Vol.52-30, pp.70-75.

(6) Hong, Y. (2016). Intelligent Detection Method of Fruit Based on Improved SSIM Algorithm. Advance Journal of Food Science and Technology, Vol.10(4), pp. 309-312.

(7) Real-Moreno, O., Rodríguez-Quiñonez, J. C., Flores-Fuentes, W., Sergiyenko, O., Miranda-Vega, J. E., Trujillo-Hernández, G., & Hernández-Balbuena, D. (2024). Camera Calibration Method Through Multivariate Quadratic Regression for Depth Estimation on a Stereo Vision System. Optics and Lasers in Engineering, Vol.174, Nov 2023.

(8) Cong, P., Li, S., Zhou, J., Lv, K., & Feng, H. (2023). Research on Instance Segmentation Algorithm of Greenhouse Sweet Pepper Detection Based on Improved Mask RCNN. Agronomy, MDPI, Vol.13, Jan 2023.

(9) Cong, P., Li, S., Zhou, J., Lv, K., & Feng, H. (2023). Research on Instance Segmentation Algorithm of Greenhouse Sweet Pepper Detection Based on Improved Mask RCNN. Agronomy, MDPI, Vol.13, Jan 2023.

(10) Yu, Y., Zhang, K., Yang, L., Zhang, D. (2019). Fruit Detection for Strawberry Harvesting Robot in Non-Structural Environment Based on Mask-RCNN. Computers and Electronics in Agriculture, Vol. 163, June 2019.

(11) Burnham, J., Hardy, J., Meadors, K. (1997). Image Processing Group: Comparison of Edge Detection Algorithms - Comparison of the Roberts, Sobel, Robinson, Canny, and Hough Image Detection Algorithms. MS State DSP Conference.

(12) Burnham, J., Hardy, J., Meadors, K. (1997). Image Processing Group: Comparison of Edge Detection Algorithms - Comparison of the Roberts, Sobel, Robinson, Canny, and Hough Image Detection Algorithms. MS State DSP Conference.

(13) Prewitt, J. M. S. (1970). Object Enhancement and Extraction. Picture Processing and Psychopictorics, pp.75-149.

(14) Wang, X. (2007). Laplacian Operator-Based Edge Detectors. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.29(5), May 2007.

(15) Canny, J. (1986). A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8(6), pp. 679-698

Nov 1986.
(16)] Hong, Y. (2016). Intelligent Detection Method of Fruit Based on Improved SSIM Algorithm. Advance Journal of Food Science and Technology, Vol.10(4), pp. 309-312.
(17) Darvatkar, S., & Bhandari, S. U. (2017). Implementation of Barrel Distortion Correction on FPGA. IEEE, 2017.
(18) Lee, M., Kim, H., & Paik, J. (2019). Correction of Barrel Distortion in Fisheye Lens Images Using Image-Based Estimation of Distortion Parameters. IEEE Access, Vol.7, pp. 45723-45733, Apr 2019.
(19) Kim, T.-H. (2018). An Efficient Barrel Distortion Correction Processor for Bayer Pattern Images. IEEE Access, Vol.6, pp. 28239-28248.
(20) Optris infrared measurements. Optris Xi 400 TECHNICAL DATA.
(21) Intel Corporation. (2020). Intel RealSense Product Family D400 Series Datasheet.