Chili Object Detection with Automatic Labeling and Mask R-CNN Model

Khine Zin Thet
University of Computer Studies, Yangon, Myanmar
khinezinthet@ucsy.edu.mm
Win Lelt Lelt Phyu
University of Computer Studies, Yangon, Myanmar
wllphyu@ucsy.edu.mm

Abstract

Abstract: To reduce labor-intensive work, we need to adopt autonomous robots in the agricultural sector. The process of traditional chili fruit harvesting requires a workforce and needs to improve for grading due to human eyes are sometime see the errors. In addition, the characteristics of chili fruits are not the same other type of fruits due to its sizes, variations, texture and its localization from the plant. To solve this issue, we conducted chili object detection using an automatic labeling algorithm and the Mask R-CNN model to localize and classify chili fruit variations. An automatic labeling algorithm is proposed for effectively exporting feature JSON files instead of pre-trained images using image annotation tools with manual processes. In our experimental results, automatic labeling detected chili objects with polygon points, and an accuracy rate of 85% was obtained with 4200 training images and 1800 testing images. The chili detection accuracy rate of 95% is tested with the Mask R-CNN model, and 2100 training images, 2100 validation images, and 1800 testing images are used in this paper.

***Keywords*: Object detection, Image Classification, Internet of Things, Image Labeling, Convolutional Neural Network, Mask R-CNN.**

1. Introduction

Agriculture has played a major role in the development and growth of societies by providing the food necessary to maintain and expand the population [6]. Nevertheless, As the human population grows, this activity faces many new problems and obstacles [8]. Demand for increased production puts significant pressure on farmers to achieve higher yield quotas, with many factors undermining this objective. Artificial Intelligence (AI) models are emerging as essential tools for farmers, facilitating crop identification and classification, and providing solutions to many challenges. When properly adjusted, AI models equipped with computer vision have the potential to reduce identification errors and improve productivity during harvest. These models can be integrated into robotic devices that can be operated automatically or remotely. The technique works particularly well in fields with a diverse array of crop species, and can accurately distinguish between fruits and leaves, a task that can be particularly difficult in crops such as chili due to their similar shapes. Different classification techniques were used when it came to fruits.

Machine Learning Deep learning and Computer Vision have gained widespread acceptance due to their commendable performance in image recognition when applied to pattern recognition. Of note are the recent advances in deep learning and Convolutional Neural Networks (CNN), which have led to the development of fast models capable of multidimensional detection and classification. A compelling illustration of their effectiveness can be found in [2], where various CNN models were subjected to rigorous tests for grape detection. Currently, traditional deep learning and machine learning algorithms solve critical tasks such as crop health monitoring [12], disease identification [1], weed detection [10]. It is recognized for the need for a wide range of human skills and labor. A few deep learning models are applied to object detection and generating neural networks face issues such as large model structures, slow training time and low accuracy.

In the view of development and needs of future deep learning technology, IoT, data analysis and smart technology. This research explored the predictions and relevance of various technologies. By conceptualizing smart image recognition and IoT as one entity, this study enabled computers to perform automated harvesting of fruits or targets through natural vision and device communication achieved with wireless communication. Machines that use low-cost microprocessors insert in hardware for capturing images in the work area can employ network models that have been pretrained using deep learning installed in server machines to improve the accuracy and speed of image recognition.

2. Related Works

In this section, a review of image classifications, detection different approaches are discussed. According to a survey, fruit detection and classification methods are still needed to develop.

2.1. *Image classifications*

Image classification is a basic operation in computer vision that classifies images into predefined classes or labels. In [7] is to classify the fruit by using CNN and RNN, LSTM based on optimum and labelled features. As an accuracy, this classification method is better than the SVM, FFNN, and ANFIS current image classification algorithms. This system needs to solve the issue of limitations in the form of contrast development and edge recognition.

This proposed [13] has developed an automated fruit using conventional augmentation techniques and deep learning techniques YOLOv3, VGG16 and ResNet50 models, in which create two databases, one for FIDS-30 dataset of 30 classes, one for custom dataset of 8 categories of fruits. This system shows the results of 86% and 85% accuracies on public dataset and 99% accuracy with ResNet50 and 98% accuracy with the VGG16 model on the custom dataset, need to improve detection and classification accuracy and reduce memory usage. In this [3] examined the performance of classification models for maturity status classification of papaya fruits i.e. VGG19 based on transfer learning approach achieved 100% accuracy which is 6% more than machine learning approach achieved.

2.2. *Image Segmentation and Object Detection*

Manual detection of fruits is a time consuming and strenuous task for the farmers. [15] This paper uses image processing and machine learning techniques for efficient detection of diseases in chili leaves. A YOLO-chili object detection algorithm for chili detection is [5] in the paper. It combines the sensitivity technique and the overall prediction concept to increase the model's expressive detection capabilities for occluded and small target peppers. In that proposed system by combining of three mechanisms to improve the algorithm's long-distance recognition ability and reduce the interference from redundant objects. Showing the results an average precision (AP) value of 93.11% for chili pepper detection, with an accuracy rate of 93.51% and a recall rate of 92.55% using custom dataset. YOLO-chili enables accurate and real-time chili detection in background clutter environments.

In [14] the resultant parameters mean Average Precision (mAP) for all three pest classes is 98.6% for YOLOv5 and 86.1% for YOLOv7. The YOLOv5s detector shows the superior performance detector in well-balanced multi-class pest type datasets compared to the YOLOv7 pest classification, with a 12.5% improvement.

3. Research Methodology

This section describes the detailed methodology required for the study and provides a detailed explanation of the process and approaches that have been used.

3.1. *Image Acquisition and Preprocessing*

Image Acquisition is the process of capturing images from various devices. For testing the auto labeling and Mask R-CNN model, a dataset was built using the image of chili plants. The images are of different climates; different lighting conditions, various angles, and time-lapse videos are converted into image frames. The number of images used in this study was 4200. These images are randomly divided into training dataset and validation dataset. In these datasets, the training datasets are used for the original images during model training and the validation datasets are used for the evaluation of the model performance after training.

3.2. *Pi Camera and Raspberry Pi Processor*

Real-time object detection requires fast and accurate processing, and achieving real-time speeds on limited high-performance systems is a challenging task. Raspberry Pi is a small processor that provides image processing capabilities and facilitates various programming options and various navigations. Connect seamlessly with peripherals such as sensors and actuators such as Pi cameras and IP cameras. In this proposed system, PI cameras capture videos of tiny green chili plants. We use a model that compares images of chili and chili trained in the Raspberry Pi's memory, and if they are the same, displays the chili to process the harvest. With the help of a neural network, the Pi circuit uses a camera to detect chili.

3.3. *Model Architecture*

Mask Region-based Convolutional Neural Network (Mask R-CNN) is a common choice for target detection tasks, including small object detection, due to many features. Mask R-CNN is effective for small object detection like chili due to some facts. Mask R-CNN is a sophisticated model that extends the faster RCNN for segmentation tasks. It not only detects target object but also generates a segmentation mask. Here's a detailed architecture of Mask R-CNN:

3.3.1. Backbone Network

This network is responsible for extracting the feature from the input image. Typically used by deep convolutional neural networks such as ResNet (ResNet-50, ResNet-101). These network processes generate a feature map from the input image; It is a spatial representation of the input image that is rich in information about the features within it.
ResNet50: ResNet50 contains 50 layers. Depth of REsNet50 is a structure including residual block. Building Blocks includes residual blocks with varying numbers of convolutional layers.
ResNet100 has 100 layers, it is deeper than ResNet50. Like ResNet50, it uses residual blocks but with a greater depth, typically involving more layers within each block or more blocks in the overall architecture. The benefits of deeper architectures like ResNet100 can potentially capture more complex features and representations in data, which may lead to improved performance on challenging datasets or tasks requiring more intricate feature extraction.
Both ResNet50 and ResNet100 are powerful architectures in the ResNet's effectiveness in deep learning tasks. ResNet100 extends the depth of ResNet50, potentially offering improved feature learning capabilities, but it requires more computational resources.

3.3.2. Feature Pyramid Network (FPN)

Mask R-CNN employs a Feature Pyramid Network (FPN) to handle the input image at different scales more effectively. FPN combines high-resolution, low-semantic feature maps with low-resolution, high-semantic feature maps to build a pyramid of multiscale feature maps. This is useful for detecting objects of different sizes.

3.3.3. Region Proposal Network (RPN)

RPN is an important component that suggests object region candidates (proposals). Slide a small network on the functional map output by Backbone + FPN. RPN generates multiple anchor boxes with different scales and aspect ratios for each location on the feature map. RPN then predicts the object score (the probability that the object exists) and the coordinates of each anchor box.

3.3.4. ROI Align

RoI Align is an improvement over RoI Pooling that addresses quantization issues. In RoI Pooling, features may shift due to rounding operations. RoI Align uses bilinear interpolation to calculate accurate values for features at fractional locations and maintains spatial alignment between the input image and feature map, which is important for mask prediction.

3.3.4. Head Network

Once regions of interest (RoIs) are extracted using RoI Align, they are fed to three parallel head networks. Classification head, this branch classifies the RoI into one of the object categories or backgrounds. It uses fully connected layers. Bounding box regression head, this branch adjusts the bounding box coordinates of each RoI. It also uses fully connected layers to predict adjustments needed to improve the accuracy of bounding box position and size. Mask prediction head, the mask prediction head is a fully convolutional network (FCN), which outputs a binary mask for each RoI. Unlike classification and bounding box heads that use FC layers, mask heads use convolutional layers to maintain the spatial structure of the RoI.

3.4. *Automatic Labeling*

Automatic labeling is a crucial process in computer vision especially in the image pre-training phase which labels the specific objects to annotate with labels or metadata to save manual pre-training tasks. Automatic labeling process has revolutionized the image analysis of large datasets in agriculture business. This method has significantly reduced not only the needs of machine learning experts but also manual tasks of annotation datasets. Automatic labeling is very popular in image pre-processing tasks such as object detection, image classification and segmentation. There are various methods which are leveraging algorithms, pre-trained models, and machine learning methodologies.

In our image analysis research works, global thresholding, morphological operations, connected component labelling and edge detection are implemented and features are exported as JSON file.

Global thresholding works well in the significant intensity difference between foregrounds and backgrounds. Global thresholding has selected a single threshold value and applied it to the entire image. The intensity of grayscale image values are compared to this threshold. The threshold value 0.3255 is used in automatic labelling of our experiments.

Morphological operations focus on the structure or objects within an image. Noise removal, shape extraction, and object segmentation processes are solved by morphological operations. In this shape extraction, morphological operations are used in the processes of boundary extraction and separation of connected objects.

The exported features files are used as input Mask R-CNN model and detail algorithm is presented in section 4. Manual labelling can be extremely time-consuming and labor-intensive, especially for large datasets, while automatic methods can process thousands of images quickly and consistently. Data labelling is the process of training a model on a limited labelled dataset which is later used for labelling new sets of data. The automatic labelling helps to achieve more accurate results and minimize human errors.

4. System Model

The main objective of the system model is to detect small fruits with Mask R-CNN which has three key stages. The first stage is the data acquisition in which chili farm videos are collected from Pi camera in different views. The second one is automatic labelling for pre-training images with the proposed algorithm. The third one is Mask R-CNN detection which significantly improved the accuracy of chili object detection with RestNet-101 backbone for flexible feature extraction to obtain the precise localization. Fig. 1 shows the system model, and three stages are discussed in section 4.1, 4.2 and 4.3.

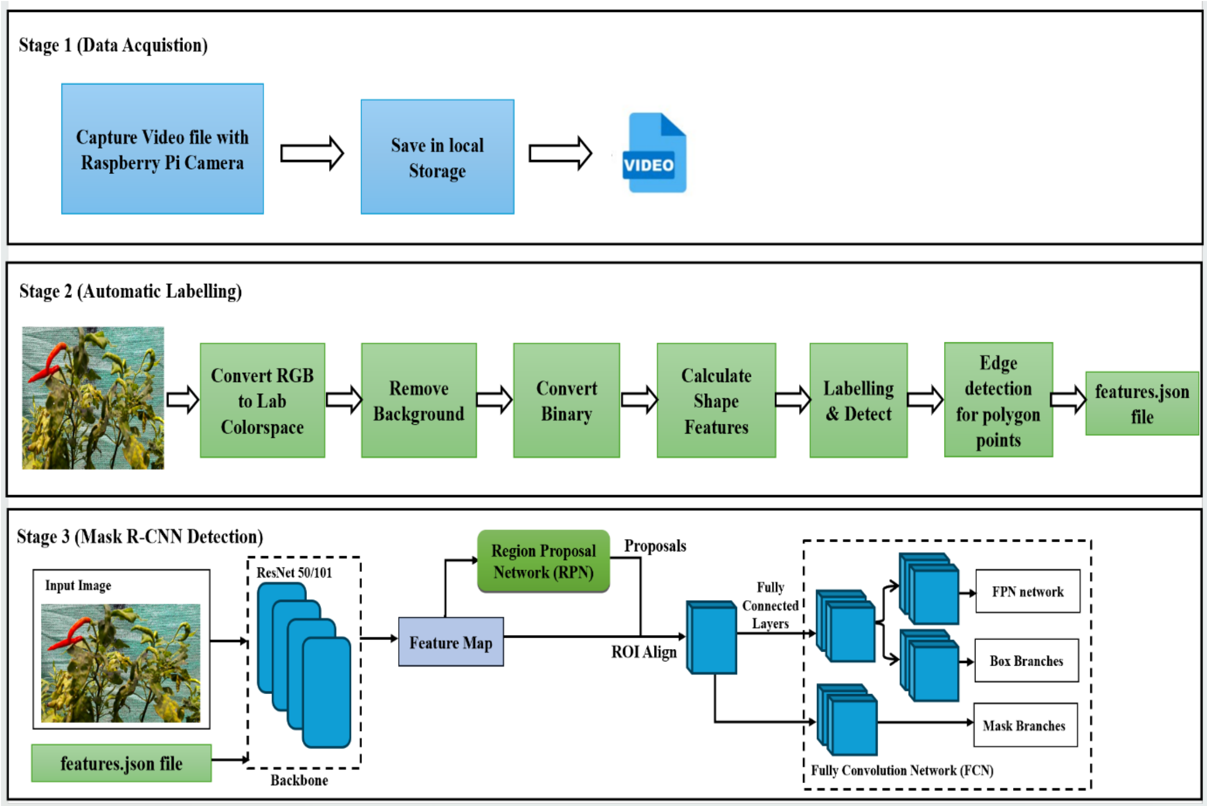


Fig. 1. System Model.

4.1. *Data Acquisition*

Data acquisition of this research work is automated collected from Pi cameras with resolution 1280X720 with 96 dpi. Capture videos 100 files are used, and 6000 frames (images) are extracted which are defined 2100 images for training data, 2100 images for validation data and 1800 images for testing data. Fig. 2 shows the sample of our chili farm. One feet distance images Fig. 2 (a) are captured from one foot far to chili plant and two feet distance images Fig. 2 (b) are from two feet. Two feet distance images are better focused and chili clarity to get good detection accuracy.

Dataset	Videos	Images
Training	35	2100
Validation	35	2100
Testing	30	1800

Table 1. Quantities of chili images in datasets.

(a) (b)

Fig. 2. Chili Plants Chili Image Sample: (a) one feet distance and (b) two feet distance.

4.2. *Automatic Labeling Algorithm*

Input RGB images are converted to Lab color space. The LAB color space is a three-dimensional color space, and they have L (Lightness) which represents the lightness of the color, ranging from 0 (black) to 100 (white), a which represents the position between red/magenta and green, b which represents the position between yellow and blue.

The conversion to Lab color space from RGB has two steps: RGB to XYZ color space with Eq. (1) and then XYZ to Lab color space with Eq. (2).

In this algorithm, the equations 1-12 are used and automatic labeling algorithm is as follow:

RGB to XYZ

$$R_n = \frac{R}{255}, G_n = \frac{G}{255}, B_n = \frac{B}{255}. \tag{1}$$

$$R_s = \begin{cases} R_n = \frac{R}{255} & if\ R_n \leq 0.04045 \\ (\frac{R_n+0.055}{1.055})^{2.4}, & f\ R_n > 0.04045 \end{cases}. \tag{2}$$

$$G_s = \begin{cases} G_n = \frac{G}{255} & if\ G_n \leq 0.04045 \\ (\frac{G_n+0.055}{1.055})^{2.4}, & f\ G_n > 0.04045 \end{cases}. \tag{3}$$

$$B_s = \begin{cases} B_n = \frac{GB}{255} & if\ B_n \leq 0.04045 \\ (\frac{B_n+0.055}{1.055})^{2.4}, & if\ B_n > 0.04045 \end{cases} \tag{4}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124564 & 0.3575761 & 0.1804375 \\ 0.2126729 & 0.7151522 & 0.0721750 \\ 0.0193339 & 0.1191920 & 0.9593941 \end{bmatrix} \begin{bmatrix} R_s \\ G_s \\ B_s \end{bmatrix}. \tag{5}$$

XYZ to Lab

$$X_n = \frac{X}{X_{ref}}, Y_n = \frac{Y}{Y_{ref}}, Z_n = \frac{Z}{Z_{ref}}. \tag{6}$$

where $X_{ref}, Y_{ref}, Z_{ref}$ are the reference white values.

$$f(t) = \begin{cases} t^{\frac{1}{3}} & if\ t > (\frac{29}{6})^3 \\ \frac{1}{3}(\frac{29}{6})^2 t + \frac{4}{29}, & if\ t \leq (\frac{29}{6})^3 \end{cases} \tag{7}$$

$$L = 116. f(Y_n) - 16. \tag{8}$$

$$a = 500. (f(X_n) - f(Y_n)). \tag{9}$$

$$b = 200. f(Y_n) - f(Z_n)) \tag{10}$$

$G(x, y)$ is grayscale matrix for Lab color space image and is solved by Eq. (11). The maximum value for L is 99.26 and minimum value for L is 0, maximum value for a is 67.7233 and minimum value for a is -24.4641, maximum value for b is 68.7045 and minimum value for b is -11.2673.

$$G(x, y) = \frac{I(x,) - minvalue}{maxvalue - minvalue} \tag{11}$$

Binarization Eq. (12) is used for converting binary image where T_2 is the binary threshold value and is T_2 0.3255.

$$\mathrm{BI(x, y)} = \begin{cases} 1 & \text{if GrayI(x, y)} > \mathrm{T_2} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

Shape features are calculated using skimage.measure module which measures area, perimeter, centroid, bounding box, major axis length, minor axis length, orientation and convex area.

Step 1: Convert RGB to Lab Color Space Lab_I = Convert_RGB2Lab(Input_I) Step 2: Remove Background BG_removed_grayI = Removed_BG(Lab_I) Step 3: Convert Binary B_I= Convert_Binary(BG_removed_grayI) Step 4: Calculate Shape Features Shape_FeaturesList = ExtractShape(B_I) Step 5: Labeling and Detect Object LabeledList = Labeledfun(B_I) Step 6: Edge Detection for Polygon Points BoundarypointsList = EdgeDetection(LabeledList) Step 7: Labeling and Detect Object LabeledList = Labeledfun(B_I) Step 8: Edge Detection for Polygon Points BoundarypointsList = EdgeDetection(LabeledList) Step 9: Write feature.json

Table 2. Automatic Labeling Algorithm

Eight connected pixels are checked for labeling to detect chili objects. Extracted labeled images are found boundary points to convert polygon contour shape and feature.Json file is exported for stage 3.

4.3. *Mask R-CNN Detection*

In Mask R-CNN model, the following configuration parameters are used. The number of training steps for each epoch is configured 1000 and validation stats to get better accuracy is 50. RestNet101 is the backbone of this model with 101 layers which has initial layer: residual blocks (stage 1: 3 residual blocks, stage 2: 4 residual blocks, stage 3: 23 residual blocks, stage 4: 3 residual blocks) and output layers. The output classes are fruits and background. The strides of each layer of the FPN pyramid are [4,8,16,32,64] and other parameters are described in table 2.

Parameter	**Value**
STEPS_PER_EPOCH	1000
BACKBONE	resnet101
BACKBONE_STRIDES	[4, 8, 16, 32, 64]
FPN_CLASSIF_FC_LAYERS_SIZE	1024
RPN_ANCHOR_RATIOS	[0.5, 1, 2]
RPN_ANCHOR_STRIDE	1
RPN_NMS_THRESHOLD	0.7
POST_NMS_ROIS_TRAINING	2000
POST_NMS_ROIS_INFERENCE	1000
MINI_MASK_SHAPE	(56, 56)
TRAIN_ROIS_PER_IMAGE	200
ROI_POSITIVE_RATIO	0.33
POOL_SIZE	7
MASK_POOL_SIZE	14
MAX_GT_INSTANCES	100
DETECTION_MAX_INSTANCES	100
DETECTION_MIN_CONFIDENCE	0.7
DETECTION_NMS_THRESHOLD	0.3
LEARNING_RATE	0.001
LEARNING_MOMENTUM	0.9
WEIGHT_DECAY	0.0001
GRADIENT_CLIP_NORM	5.0

Table 3. Mask R-CNN with RestNet101 Backbone Model Configuration

5. Experimental Results

Experiment results are discussed with two sections: 5.1 for automatic labeling steps and 5.2 for Mask R-CNN chili detection.

5.1. *Automatic Labeling*

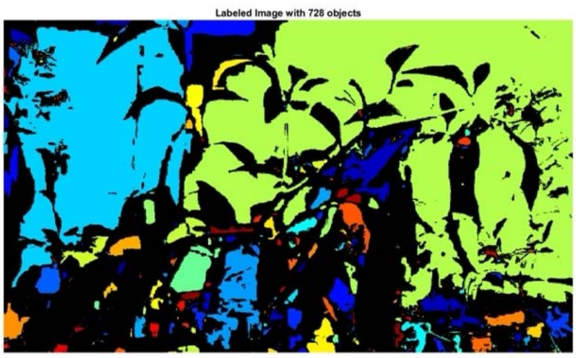
The experimental results of automatic labeling processes of the original image (a), Lab color image (b), background removed image (c), binary image (d) and labeled image 728 objects (e) are shown in Fig. 3. In our experimental results, automatic labeling is detected chili objects with polygon points and accuracy rate 85% is obtained with 4200 training images and 1800 testing images.

(a) Original Image.

(b) Lab Color Space Image.

(c) Background Removed Image.

(d) Binary Image.



(e) Labeled Image with 728 objects.

(f) Polygon Line and Object Boundary.

Fig. 3. Results of Automatic Labeling.

5.2. *Mask R-CNN*

Input image resolution 1280X720 images are used in this experiment as in Fig. 4. Aspect ratio is preserved, and images are resized to 1024X1024 with zero padding at the top, bottom, left and right matrix. The RPN target match is 1 (positive anchors), -1 (negative anchors) and 0 (neutral anchors).

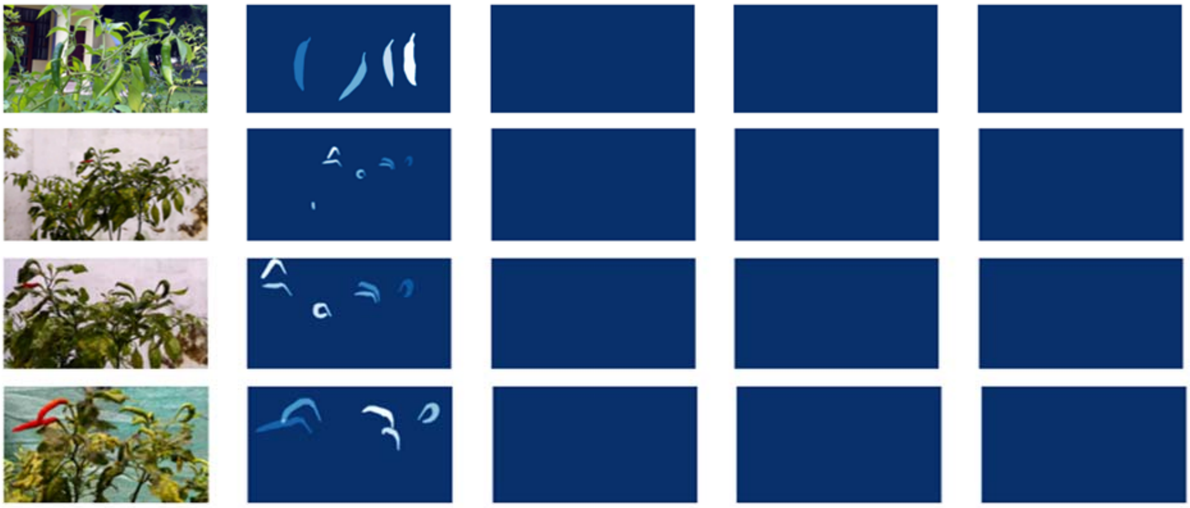
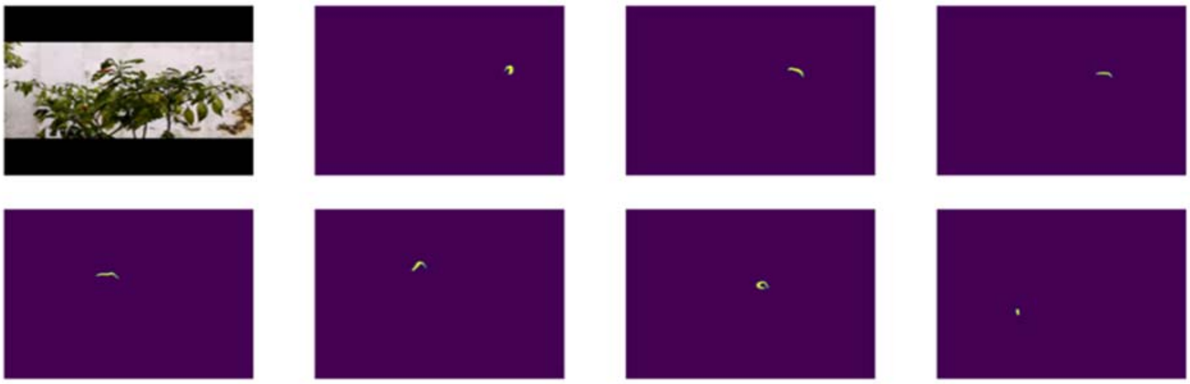
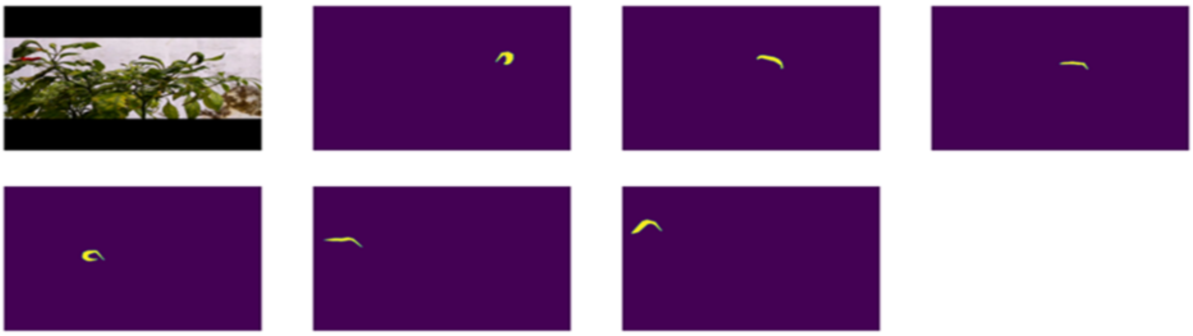


Fig. 4. Original Images (1280X720) and Masks.

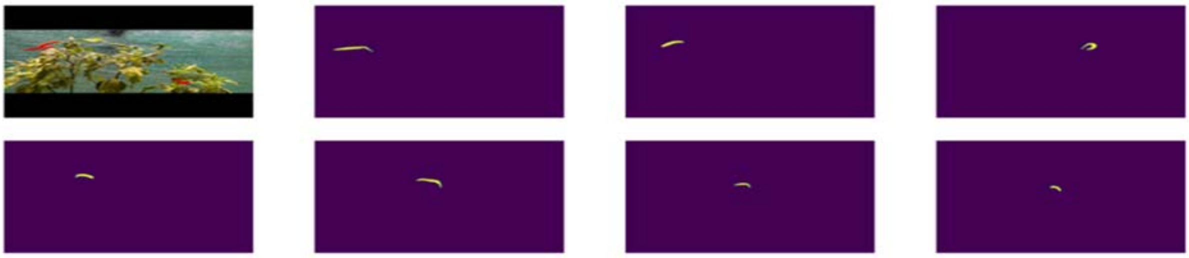
(a)



(b)



(c)



(d)
Fig. 5. 1024X1024 Image and Masks

Fig. 6. Not Refinement (Dotted lines) and Refined (Solid lines) Image

Fig. 5. (a,b,c,d) shows the resized image and masks. The dotted rectangular boxes are not refinement outputs and solid rectangular boxes are represented as refined outputs and shown in Fig. 6. Level 0 to 4 Anchors, feature maps are shown in table 4 Fig. 7 shows the detected mask chili image.

Anchors	Feature Map	Dimension
Level 0	196608	[256 256]
Level 1	49152	[128 128]
Level 2	12288	[64 64]
Level 3	3072	[32 32]
Level 4	768	[16 16]

Table 4. Feature Map of Level 0-4.

Fig. 7. Detected Mask Chili Image.

5.3. *Model Performance Evaluation*

In this proposed system, the performance evaluation such as Precision Eq. (13), Recall Eq. (14), mAP Eq. (15), F1 score Eq. (16) of the trained Mask RCNN model. The equations are as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{13}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{14}$$

$$\text{mAP} = \frac{1}{c}\sum_{K=i}^{N} \text{P(k)}\Delta\text{R(k)} \tag{15}$$

$$\text{F} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} \tag{16}$$

Where TP = the number of correct identified two varieties of chili targets, FP = the number of misidentified backgrounds as chili targets, FN = the number of unidentified chili targets, C = the number of chili targets categories, N = the number of IOU thresholds, K = the IOU threshold, P(k) = the precision, and R(k) = the recall. The chili detects an accuracy rate of 95% is tested with Mask R-CNN model and 2100 training images, 2100 validation images and 1800 testing images are used in this paper. Our model's performance achieved 93.7% precision, 94.8% recall, and 94.2% F1 score when detection of the chili objects.

6. Conclusion

In this proposed system used a Mask R-CNN deep learning method and automatic labeling detected chili objects with polygon points on IoT technology to improve a remote crop picking system. We make models for labelling and detection. The classification of chili accuracy rate is 95% and automatic labeling is detected chili accuracy rate 85% respectively. Further process is to compare the model of analysis chili detection time and localization for picking chili. This study contributes the application of deep learning method and automated labeling on collected images from Raspberry Pi camera video like small chili for smart image recognition.

Conflicts of Interest

The authors have no conflicts of interest to declare.

References

[1] Ahmad, A., Saraswat, D., & El Gamal, A. (2023). A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools. Smart Agricultural Technology, 3, 2772–3775.
[2] Badeka, E., Karapatzak, E., Karampatea, A., Bouloumpasi, E., Kalathas, I., Lytridis, C., Tziolas, E., Tsakalidou, V. N., & Kaburlasos, V. G. (2023). A Deep Learning Approach for Precision Viticulture, Assessing Grape Maturity via YOLOv7 Sensors.
[3] Behera, S. K., Rath, A. K., & Sethy, P. K. (2021). Maturity status classification of papaya fruits based on machine learning and transfer learning approach. Information Processing in Agriculture, 8(2), 244–250.
[4] C. T. C. Doloriel and R. D. Cajote, "Improving the Detection of Small Oriented Objects in Aerial Images," 2023 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW), Waikoloa, HI, USA, 2023, pp. 176-185.
[5] Chen, Hailin & Zhang, Ruofan & Peng, Jialiang & Peng, Hao & Wang, Yi & Jiang, Ping. (2024). YOLO-Chili: An Efficient Lightweight Network Model for Localization of Pepper Picking in Complex Environments. Applied Sciences. 14. 5524. 10.3390/app14135524.
[6] Godfray, H. C. J., Crute, I. R., Haddad, L., Muir, J. F., Nisbett, N., Lawrence, D., Pretty, J., Robinson, S., Toulmin, C., & Whiteley, R. (2010). The future of the global food system. Philosophical Transactions of the Royal Society B: Biological Sciences, 365(1554), 2769–2777.
[7] Harmandeep Singh Gill, Baljit Singh Khehra. Fruit Image Classification using Deep Learning, 07 November 2022, PREPRINT (Version 1) available at Research Square.
[8] Hou, X., Li, J., Yin, S., & Yuan, H. (2022). Evolution of Agricultural Innovation Ecosystem in County Areas: A Life-Cycle Perspective of Cases in Hebei Province. Mathematical Problems in Engineering, 2022, 1–21.
[9] Indrabayu, Mar'atuttahirah and I. S. Areni, "Automatic Counting of Chili Ripeness on Computer Vision for Industri 4.0," 2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), Bali, Indonesia, 2019, pp. 14-18.
[10] Islam, N., Rashid, M. M., Wibowo, S., Xu, C. Y., Morshed, A., Wasimi, S. A., Moore, S., & Rahman, S. M. (2021). Early weed detection using image processing and machine learning techniques in an Australian Chilli farm. Agriculture (Switzerland).
[11] Kanaparthi, Kantha et al. "Deep Learning Methods for Detecting Chilli Pests: A Novel Performance Analysis." Ecological Engineering & Environmental Technology, vol. 25, no. 6, 2024, pp.234-254.
[12] Khalid, M., Sarfraz, M. S., Iqbal, U., Aftab, M. U., Niedbała, G., & Rauf, H. T. (2023). Real-Time Plant Health Detection Using Deep Convolutional Neural Networks. Agriculture (Switzerland), 13(2), 2–16.
[13] Mimma, Nur-E-Aznin, Ahmed, Sumon, Rahman, Tahsin, Khan, Riasat, Fruits Classification and Detection Application Using Deep Learning, Scientific Programming, 2022, 4194874, 16 pages, 2022.
[14] Moya, V., Quito, A., Pilco, A., Vásconez, J. P., & Vargas, C. (2024). Crop Detection and Maturity Classification Using a YOLOv5-Based Image Analysis. Emerging Science Journal, 8(2), 496-512.
[15] Y. Sambrani, Rajashekarappa and S. Bhairannawar, "Chili Disease Detection and Classification using Various Machine Learning Techniques," 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), Dharwad, India, 2023, pp. 1-6.

Authors Profile

Khine Zin Thet received a Master of Computer Science degree from the University of Computer Studies, Yangon in 2016. She also received her B.C.Sc. and B.C.Sc. (Hons:) from the University of Computer Studies, Yangon in 20011 and 20012, respectively. She is currently a research candidate at the University of Computer Studies, Yangon. Her research includes deep learning, Internet of Things, object detection. She can be contacted at email: khinezinthet@ucsy.edu.mm.

Dr. Win Lelt Lelt Phyu received Ph.D. in Information Technology in 2006. Currently, she is a professor at the Artificial Intelligence Lab at the University of Computer Studies, Yangon, Myanmar. She has been supervising master's thesis and Ph.D. research in machine learning and artificial intelligence. She can be contacted via email at wllphyu@ucsy.edu.mm.