Cassava stalk detection for a cassava harvesting robot based on YOLO v4 and Mask R-CNN

Thanaporn Singhpoo,[1] Khwantri Saengprachatanarug,[1,2] Seree Wongpichet,[2] Jetsada Posom,[1,2] Kanda Runapongsa Saikaew[3]

[1]Department of Agricultural Engineering, Faculty of Engineering, Khon Kaen University; [2]The Northeast Thailand Cane and Sugar Research Center (NECS), Khon Kaen University; [3]Department of Computer Engineering, Faculty of Engineering, Khon Kaen University, Thailand

Abstract

The quality of fresh cassava roots can be increased through the use of precision equipment. As a first step towards developing an automatic cassava root cutting system, this study demonstrates the use of a computer vision system with deep learning for cassava stalk detection. An RGB image of a cassava tree mounted on a cassava-pulling machine was captured, and the YOLO v4 model and two Mask R-CNN models with ResNet 101 and ResNet 50 base architectures were employed to train the weights to predict the position of the cassava stalk. One hundred test images of stalks of various shapes and sizes were used to determine the grasping point and inclination, and the results from manual annotation were compared with the predicted results. Regarding localisation, Mask R-CNN with ResNet 101 gave a significantly higher performance than the other models, with an F1 score and a mean IoU of 0.81 and 0.70, respectively. YOLO v4 showed the highest correlation for the *x*- and *y*-coordinates for the prediction of the grasping point, with values for R^2 of 0.89 and 0.53, respectively. For inclination prediction, Mask R-CNN with ResNet 101 and Mask R-CNN with ResNet 50 gave the same level of correlation, with values for R^2 of 0.50 and 0.61, respectively. These results were acceptable for use as design criteria for developing a cassava root-cutting robot.

Correspondence: Khwantri Saengprachatanarug, Department of Agricultural Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen, 40002, Thailand.
Tel.: +668.7668.9270. E-mail: khwantri@kku.ac.th

Key words: automatic harvester, cassava root, computer vision, crop detection, stalk detection.

Acknowledgements and funding: this work was supported by the Research Fund for Supporting Lecturers to Admit High Potential Students to Study and Research in His Expert Program Year 2020 [grant number 631T224]. The authors are grateful to Research and Graduate Studies, Khon Kaen University, and the Northeast Thailand Cane and Sugar Research center (NECS) , Khon Kaen University, Thailand, for providing the laboratory and equipment required for this study.

Received: 3 November 2021.
Accepted: 22 June 2022.

Journal of Agricultural Engineering 2023; LIV:1301
doi:10.4081/jae.2023.1301

Introduction

Cassava (*Manihot esculenta*) is a tropical root crop that provides nutrients and is a staple food for an estimated 800 million people worldwide. A total of 19.6 million hectares of cassava are grown in 36 countries within Africa, Asia, and South America. Moreover, high-quality cassava flour is used as a material for plywood, paper, and textiles (FAO, 2013; KURDI, 2020; OAE, 2021). Fresh roots harvested with low downtime and low amounts of trash (impurity) are required to produce this high-value product.

Unlike other field crops, the sophisticated morphology of the cassava plant means that the automatic separation of the roots and stem poses a significant challenge. This process generally consists of grasping, alignment, and cutting. The best grasping point is a specific position on the old stalk, located not too far from the first root, with sufficient strength to resist breaking when the roots are cut (Suvanapa and Wongpichet, 2014; Singhpoo *et al*., 2019). In addition, the axis of the stalk must also be perfectly aligned with the axis of the cylindrical saw during cutting to avoid high percentages of loss and trash (Mauntumkarn, 2010; Manthamkan *et al*., 2011; Chansiri and Wongpichet, 2011; Sangphanta *et al*., 2011, 2015; Vatakit *et al*., 2014). Hence, manual cutting has been the most common practice to date, making harvesting a skilled, labour-intensive, and time-consuming process (Langkapin *et al*., 2012; Vatakit *et al*., 2014). This means that the quality of the roots is easily affected by labour shortages and inexperienced workers. A cutting, grasping, and aligning robot using a computer vision system is crucial to eliminating these drawbacks.

The computer vision system in this study was used in conjunction with a cassava root-cutting robot. The robot consists of a cylinder saw and grasper mechanism; this is the optimal type of equipment, as the cutting results can satisfy the farmers' requirements. The robot can be implemented in many forms, such as a mobile cutting robot or a cutting robot on a pulling machine, as illustrated in Figure 1. In the former case, cassava plants are pulled, gathered, and dumped in a field by a pulling and gathering machine. The cutting robot is then transported to each cassava stack to cut and convey the cassava root to the truck (Figure 1A). The other option is to attach the cutting robot to the pulling machine (Figure 1B). These applications aim to minimise labour requirements while maximising working capacity under actual field conditions.

In the domain of agriculture, computer vision systems and deep learning have been used in several applications for a range of crops, such as yield estimation, growth state prediction, and harvesting point detection. Currently, research on computer vision applied to cassava harvesting is scarce; the most similar work to the present study focused on calculating the picking point on a branch for fruit harvesting. However, the use of computer vision for robotic harvesting machines for products such as apples, broccoli, strawberries, pears, kiwi, tomatoes, oranges, wine grapes, mangos, and cucumbers, has been proposed by various researchers (Blok *et al*., 2016; Font *et al*., 2014; Fu *et al*., 2020; Ganesh *et al*., 2019; Ge *et al*., 2019; Jia *et al*., 2020; Koirala *et al*., 2019; Ling *et al*., 2019; Majeed *et al*., 2019; Mao *et al*., 2020; Tian *et al*., 2019; Williams *et al*., 2019; Yu *et al*., 2019).

The most popular deep learning algorithms for crop recognition are *you only look once* (YOLO) and the *mask region convolutional neural network* (Mask R-CNN). YOLO v4 is a state-of-the-art, real-time object detection system that is swift and accurate. Although other algorithms, such as R-CNN, Fast R-CNN, and Faster R-CNN, can detect a target with high accuracy (Girshick, 2015; Ren *et al*., 2017), they have slow detection speeds and cannot produce real-time results with high image resolution. In contrast, YOLO v4 unifies the target classification and localisation stages into a regression problem and can provide much faster detection that can operate in real-time (Redmon *et al*., 2016; Redmon and Farhadi, 2018; Bochkovskiy *et al*., 2020). Nevertheless, this algorithm can only roughly calculate the target position using a bounding box and cannot accurately extract contour and shape information (Yu *et al*., 2019). Cassava stalks have widely varying sizes, irregular shapes, and individual inclination angles when hanging on the grasper (as the roots are being cut). Mask R-CNN, a state-of-the-art approach in image segmentation, can therefore be helpful as it generates a high-quality segmentation mask (He *et al*., 2020). In this work, a high-precision mask of the shape of the cassava stalk (a segmentation mask) is recognised using this method, and the optimal grasping point and inclination are then calculated from the segmentation area.

In this study, we applied YOLO v4 and Mask R-CNN to detect the grasping point of the cassava plant and its inclination after being lifted from the ground. In addition, the localisation and prediction performance of our approach was evaluated. The detection results are beneficial for setting the cutting blade alignment. This accurate detection method will accomplish low percentages of loss and trash and continuous successful cutting. Moreover, it can be used to develop a cassava root-cutting robot, reducing the cost and time involved and increasing the quality of the harvested cassava roots.

Materials and Methods

The experimental procedure used in this study consists of seven steps: image acquisition, image grouping, image annotation, model training, model testing or prediction, model evaluation, and model selection. The workflow is illustrated in Figure 2, where a dashed line represents the process from image acquisition to model training. When the weights have been calculated, they are applied to predict the images in the test set. The solid line shows the workflow from the prediction stage onwards.

Image acquisition

A variety of cassava called Kasetsart 50, the most popular variety in Thailand (OAE, 2021), was used in this study. All specimens were grown in Khon Kaen, Thailand, between October 2019 and August 2020. The age of the samples was 10 months (the average harvesting age is usually 10 to 12 months), and they were harvested under the same conditions in which the pulling machine was used.

An RGB-D Intel RealSense D435 camera was used to capture photographs of the cassava plants in the form of RGB images. The camera was fixed on a stand placed 1 m away from the cassava-pulling machine, as shown in Figure 3. A rigid frame needed to be used for the camera stand and cassava hanging point to avoid changes in depth. The camera angle was set to a constant value, and the object (cassava) plane was set parallel to the camera plane and remained unchanged. This distance of 1 m between the camera and the cassava was the minimum necessary to capture all parts of the cassava and the pulling machine. The experiment was carried out at the laboratory scale. The brightness of the images in the set used in this study varied due to the illumination from the sun in a similar way to the expected actual working conditions, as noted in Figure 1. In both sunny and cloudy conditions, 1700 cassava images were recorded at different times of the day (morning, noon, afternoon and evening). All cassava photographs were acquired as RGB images and stored in the portable network graphics (PNG) file format with a resolution of 1280×720.

Model training

Two methods trained the models: object detection (one model, YOLO v4) and segmentation [Mask R-CNN with two models: residual network (ResNet) 101 and 50]. The images were annotated *via* two software. First, the full set of 1700 images was randomly divided into two datasets: a training set (1600 images) and a test set (100 images). The labelling in the CiRA CORE program (Boonsang, 2020a, 2020b) was applied to annotate the cassava stalks in the 1600 images in the training set with a rectangular shape. These annotated images were then used to train the object detection method from the CiRA CORE program (YOLO v4 model). In general, the performance of a deep learning model improves when the amount of data is increased. Image augmentation is a technique that can be used to expand the number of existing images for the training process (Saxena, 2021) and can prevent overfitting of the model. This approach was therefore applied to augment our image set using rotation, contrast, noise, and blur to give a total of 32,000 images.

We also used another annotated software program called the visual geometry group image annotator (VIA) (Dutta and Zisserman, 2019). All images (1700) were annotated with polygon shapes to create mask images of the cassava stalks. The images were stored in the JavaScript object notation (JSON) file format. The 1600 images in the training set were randomly separated into two sub-groups, a training group (1280 images) and a validation group (320 images) and were trained using Mask R-CNN. The 100 images in the test set were used as target images in the testing step for comparison with the predicted images. The image set included images of cassava stalks of various shapes and sizes, as displayed in Figure 4.

The training platform consisted of a computer with an Intel(R) Core (TM) i7-7700 CPU, 3.60 GHz, 16.0 GB of memory, a 64-bit operating system, Windows 10 Pro, and Nvidia GeForce GTX 1060 GPU. The software tool for YOLO v4 was the CiRA CORE program, while the software tools for Mask R-CNN included CUDA 9.0, cuDNN 7.6.5 for CUDA 9, Python 3.6, Anaconda Navigator, Jupyter Notebook, and Microsoft Visual Studio 15.0.

Mask R-CNN

Before Mask R-CNN was developed, Faster R-CNN was widely used in object detection work, and the fully convolutional network (FCN) was used in the semantic segmentation field. Currently, Mask R-CNN is driven by powerful baseline systems such as the Fast/Faster R-CNN and FCN frameworks for object detection and semantic segmentation, respectively, as illustrated in Figure 5. Faster R-CNN is an extension of R-CNN and Fast R-CNN. The platform used by R-CNN for object detection consists of two stages. The first, a region proposal network (RPN), applies to bound boxes to potential objects, while the second and most essential stage, Fast R-CNN, extracts features using RoIPool from each potential box and then performs classification and bounding-box regression. The features used by both stages can be shared for faster inference. The same two steps are applied in Mask R-CNN. The first stage is an RPN, while the second is used in parallel to predict the class and box offset, and a binary mask for each RoI is also output by Mask R-CNN. This is the difference between the most recent systems. The classification depends on mask predictions and follows the principle used in Fast R-CNN to carry out bounding-box classification and regression in parallel (Abdulla, 2016; He *et al*., 2020).

YOLO

In our approach, localisation of the cassava stalk was done by the object detection method, to compare its ability with the segmentation method to select the best computer vision system for a cassava root-cutting machine.

Object detection involves locating the position of an object in an image with a bounding box and indicating its class. The YOLO model was applied in this case, as shown in Figure 6. Generally, the accuracy of a YOLO model is lower than that of R-CNN, but its computing time is faster than R-CNN, and it can achieve real-time object detection (Brownlee, 2021). The YOLO model can be applied to an image at multiple locations and scales, and the high-scoring regions of the image are considered for detection.

The model works by first splitting the input image into a grid of cells, each of which is responsible for predicting a bounding box if the centre of a bounding box falls within the cell. A bounding box is predicted by each grid cell and contains the *x*- and *y*-coordinates, width, height, and a confidence score. A class prediction is also made for each cell (Bochkovskiy *et al*., 2020; Redmon and Farhadi, 2018).

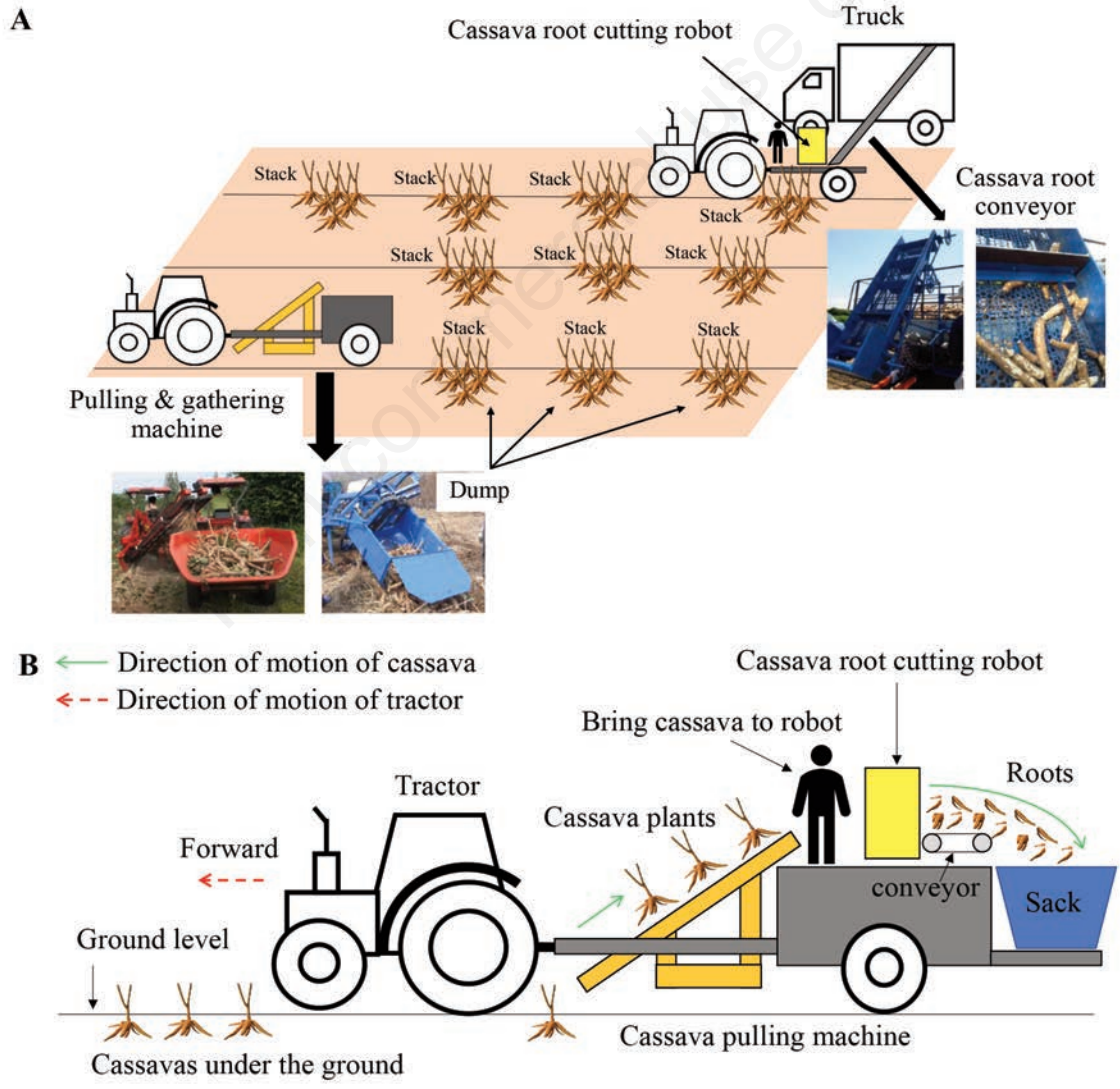

Figure 1. Operation of the system: **A**) mobile robot; **B**) robot on pulling machine.

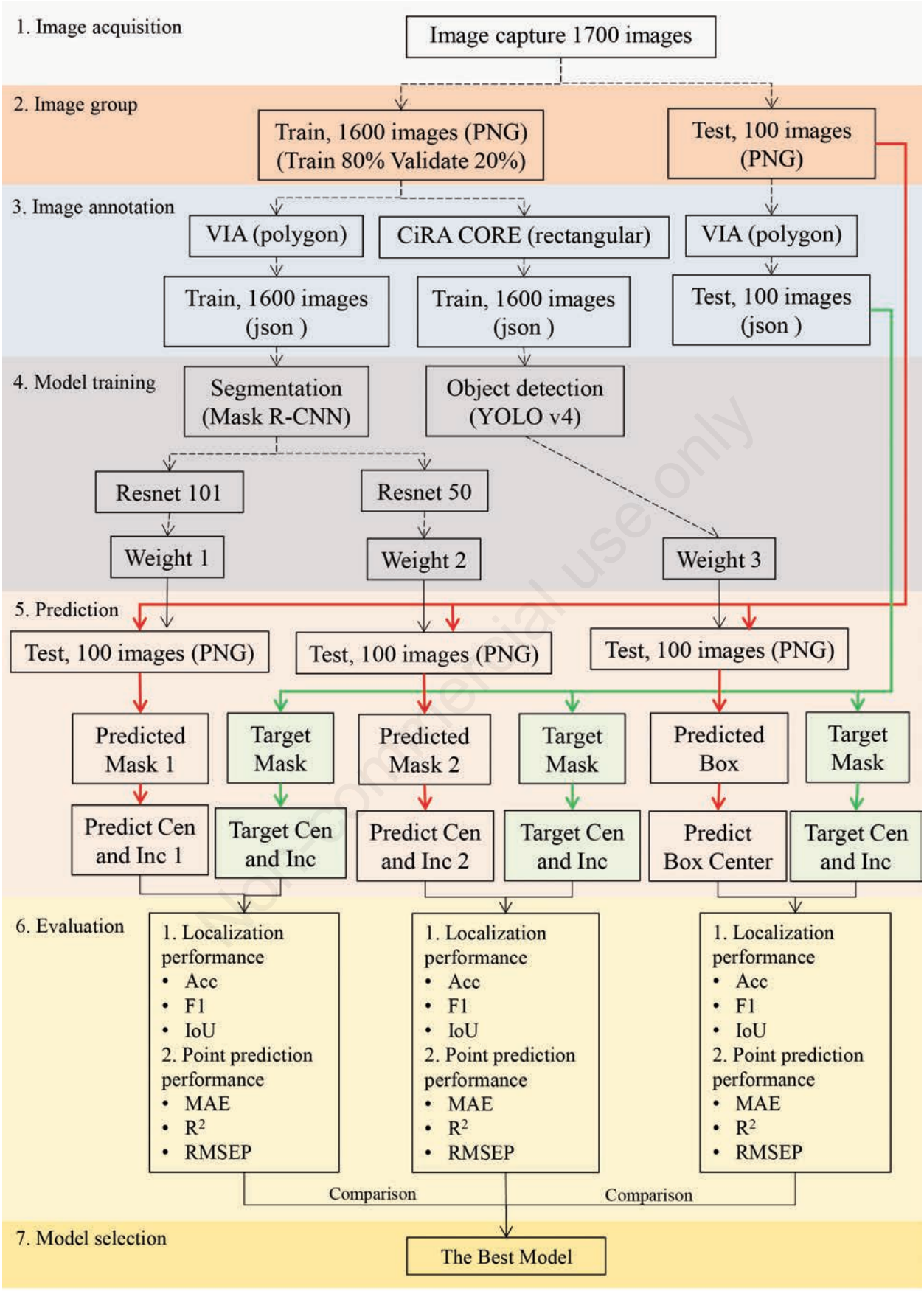

Figure 2. Experimental procedure.

Centroid and inclination prediction

The weights of the three models obtained from the training process were used to detect the region of the cassava stalk in 100 test images. For the YOLO v4 model, the centre was generated based on the predicted bounding box. The inclination was the limitation of the YOLO v4 model; it cannot generate the object inclination. Its inclination was the vertical axis of the image (with an inclination parallel to the border of the bounding box in the vertical direction). Example images showing detection results are given in Figure 7. For the Mask R-CNN model, a further calculation was needed. The model predicted the mask image of the cassava stalk region, and the contour line of the mask image was created. The centroid of the contour line was determined, as shown in Figure 8. The contours can be described simply as a curve joining all the continuous points (along the boundary) with the same colour or intensity. These contours are useful for shape analysis, object detection, and recognition (Mordvintsev and Abid, 2013). The centroid of a shape is the arithmetic mean of all the points within it. If a shape consists of n distinct points, $x_1 \ldots x_n$, then the centroid is given by:

$$c = \frac{1}{n}\sum_{i=1}^{n} x_i \quad (1)$$

In image processing and computer vision, each shape consists of a number of pixels, and the centroid is simply the weighted average of all the pixels constituting the shape (Bapat, 2018).

When the centroid has been determined in pixel coordinates, it must be converted into actual coordinates. Therefore, a chessboard marker was mounted on the pulling machine at the same depth as the cassava stalk. This marker appeared in all test images, and its centre defined the origin point. Its dimensions were 50 mm in width and height, as illustrated in Figure 9, and these dimensions were applied to convert the pixel length to the actual scene. After the centroid of the cassava stalk had been predicted, the distance to the origin point was measured in pixels and converted to the length in the actual scene in units of centimetres. This way, the length was converted to the coordinates in the actual scene.

Principal component analysis (PCA) was then used to find the inclination in the segmentation method. Finally, the line of best fit was created from the point collection (the pixels) of the mask image, and the angle between this line and the horizontal line of the object centroid was calculated as the inclination. This solution was adapted from an example given in Stack Overflow (2017).

The performance of each model was evaluated by comparing the centroid coordinate and the inclination from a predicted mask image with those of a manually annotated mask image. In addition, the region of the cassava stalk in each image was annotated, and the remainder of the image was set as the background. A comparison of cassava stalk images is shown in Figure 8.

Evaluation of cassava stalk localisation

Accuracy is a metric used to evaluate the localisation performance by simply reporting the percentage of pixels in the image that are correctly classified. However, this measure can sometimes provide misleading results when the object represented in the image is small (Jordan, 2018). Therefore, the precision, recall, and F1 score are also used to evaluate the model.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

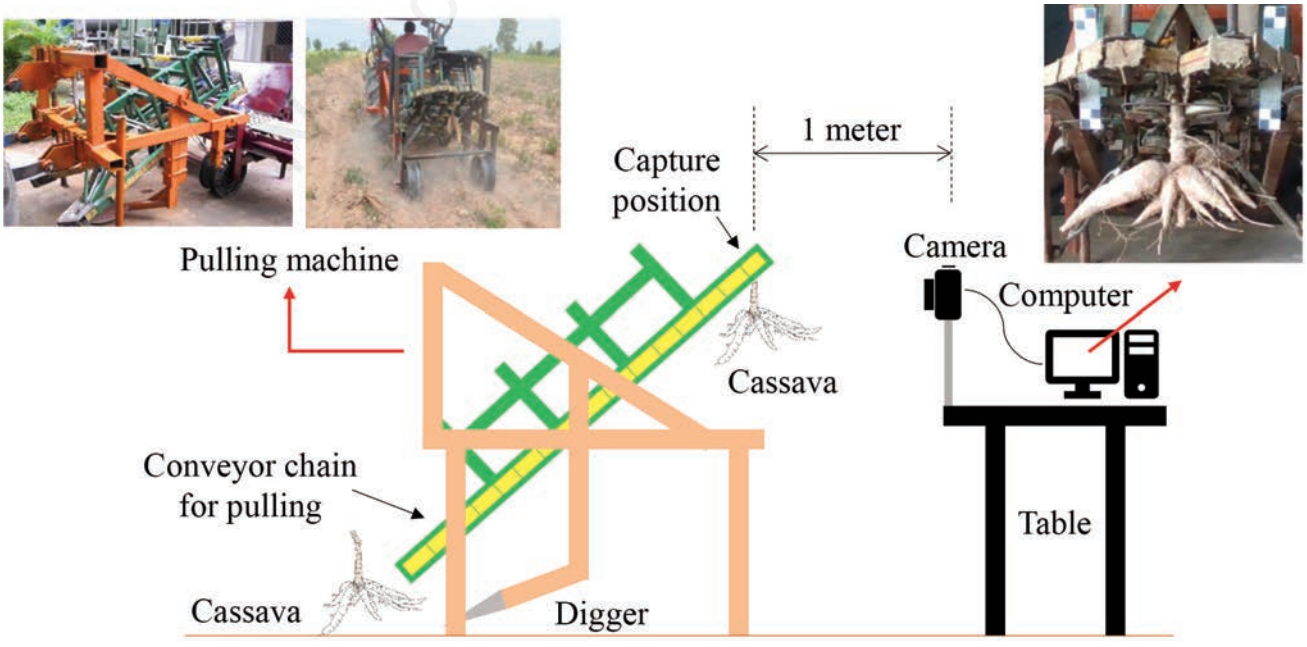

Figure 3. Schematic of image acquisition in the laboratory.

$$F1\ score = 2\times\frac{(Precision \times Recall)}{(Precision + Recall)} \quad (5)$$

where TP is the number of pixels that are correctly classified as cassava stalk, TN is the number of pixels that are correctly classified as not cassava stalk, FP is the number of pixels that are incorrectly classified as cassava stalk, and FN is the number of pixels that are incorrectly classified as not cassava stalk.

The intersection over union (*IoU*) is another method of evaluating the localisation performance of the model by calculating the intersection and union between the target area and the predicted area. In Mask R-CNN, the target area was the mask, whereas in YOLO v4, the target area was the mask identified by manual annotation, and the predicted area was the predicted bounding box.

$$IoU = \frac{P^t \cap P^p}{P^t \cup P^p} \quad (6)$$

where P^t represents the pixels labelled as cassava stalk in the target image, and P^p denotes the pixels labelled as cassava stalk in the predicted image.

Evaluation of grasping point and inclination prediction

The grasping point and inclination prediction performance were statistically evaluated based on the mean absolute error (MAE), coefficient of determination (R^2), and root mean square error of prediction (RMSEP), which can be calculated as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_{ti} - y_{pi}| \quad (7)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_{ti}-y_{pi})^2}{\sum_{i=1}^{n}(y_{ti}-\bar{y})^2} \quad (8)$$

$$RMSEP = \sqrt{\frac{\sum_{i=1}^{n}(y_{ti}-y_{pi})^2}{n}} \quad (9)$$

where y_{ti} is the measured value of the target data, y_{pi} is the predicted value, $\bar{y}$ is the average of the measured value of the target data, and n is the number of samples. R^2 denotes the proportional variance of the coordinates and inclination in the target value that was determined from the predicted value. *RMSEP* denotes the root mean square error of prediction. If R^2 is low, the model development is unnecessary. *RMSEP* represents the average uncertainty that could be expected for predicting future samples (Maraphum *et al*., 2020).

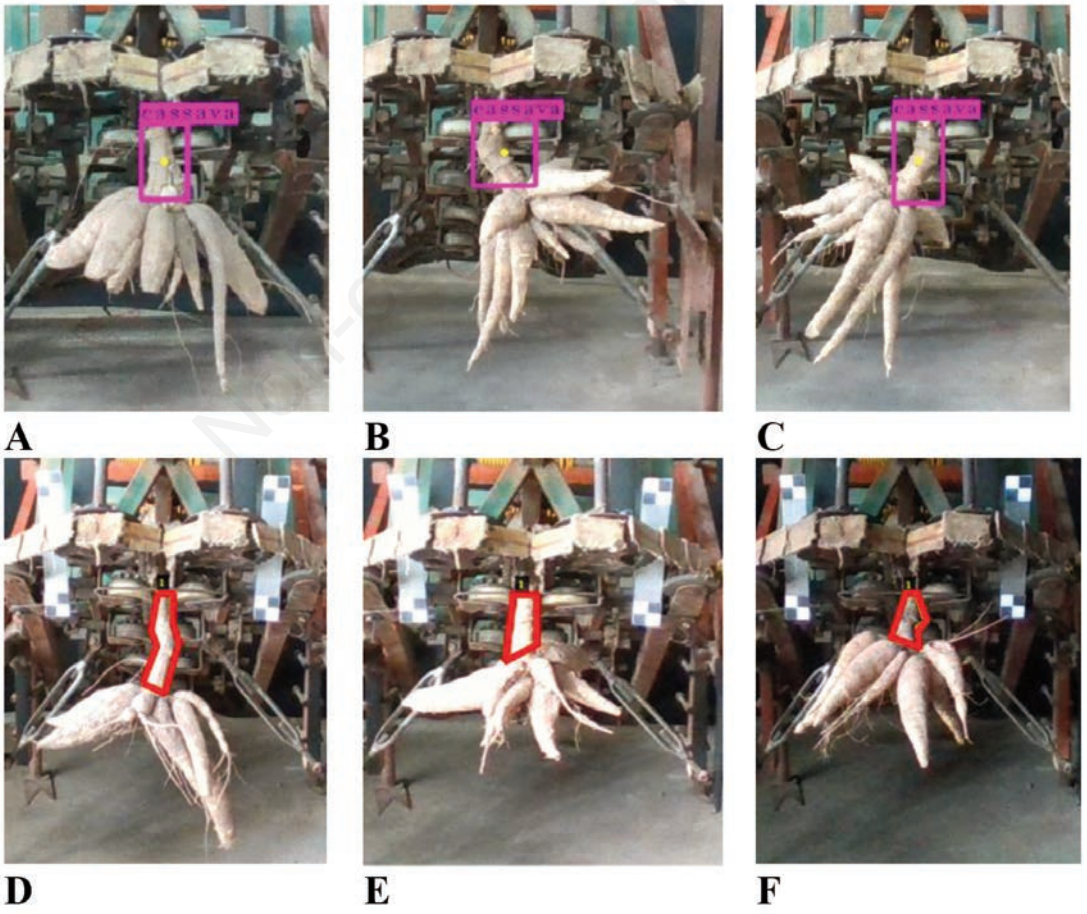

Figure 4. Cassava stalk annotation: (**A-C**) annotation in CiRA CORE program; (**D-F**) annotation in VIA.

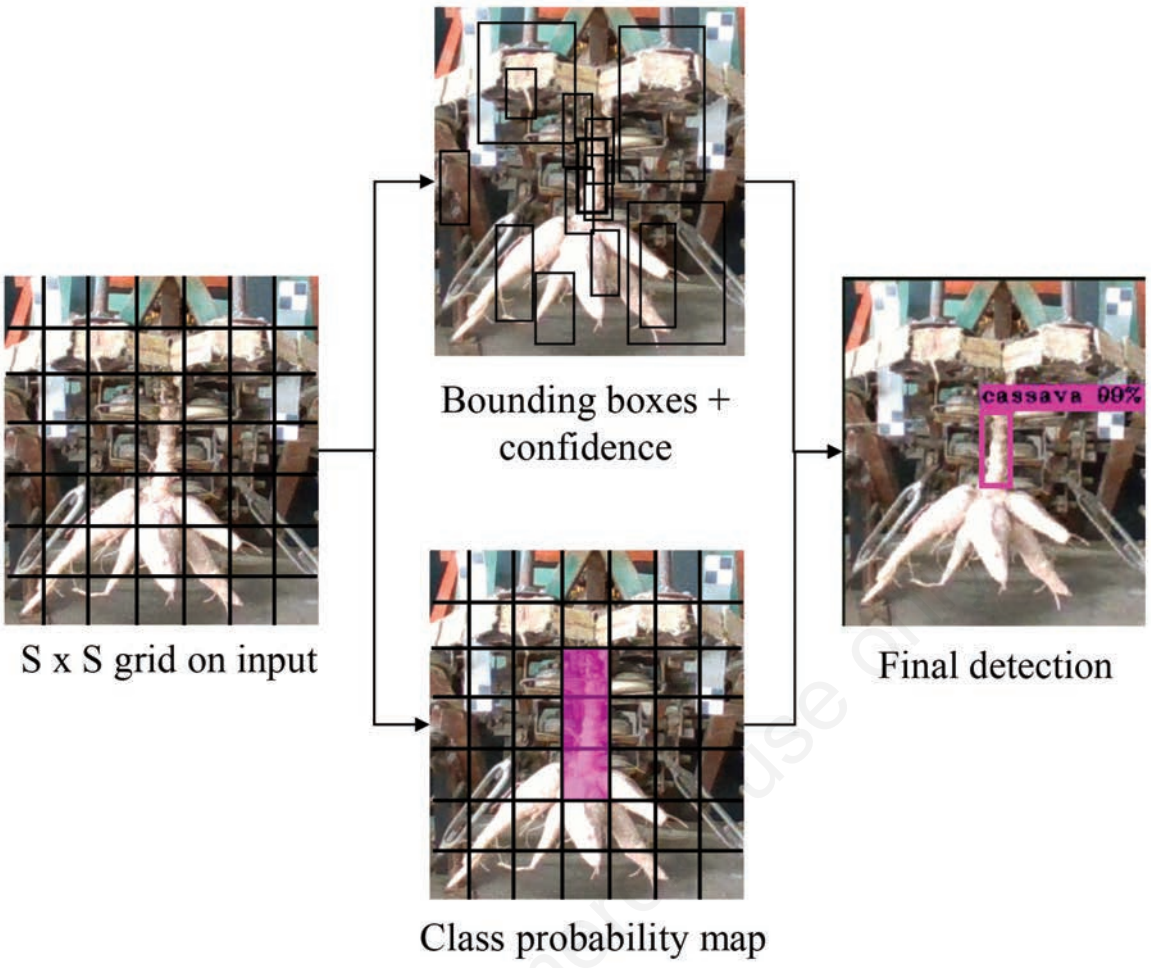

Figure 6. YOLO detection.

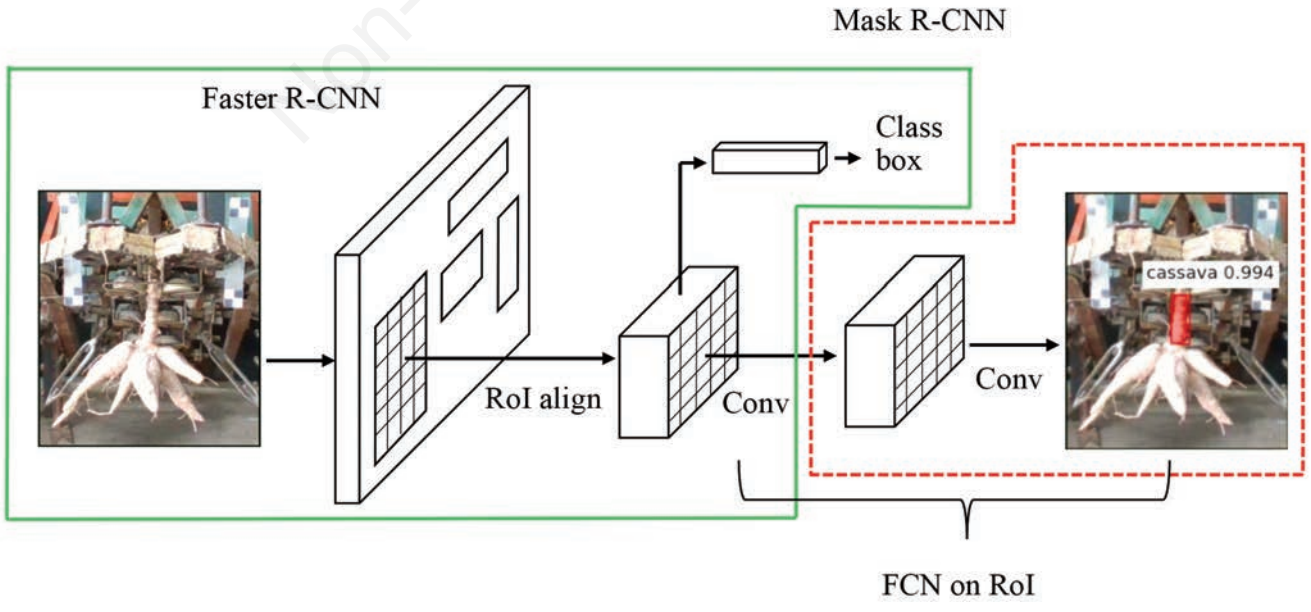

Figure 5. Structure of Mask R-CNN.

Results

Model training results

For the object detection training of YOLO v4 in the CiRA CORE platform, the parameters were set to the defaults for the program. For Mask R-CNN, ResNet 50 and 101 were used as backbone networks (Abdulla, 2016) with a feature pyramid network (FPN). The training included 1600 labelled cassava stalk images and was carried out over 3000 steps (30 epochs of 100 steps each). The batch size was two, the validation step was 50, and the learning rate was 0.001. The model loss function reached a convergence state. The loss function showed a downward trend during training, showing that the deviation in the prediction loss of the model decreased slowly by updating the loss function during the optimisation process. When the number of epochs was more than 15, the loss function values of the training and validation set decreased

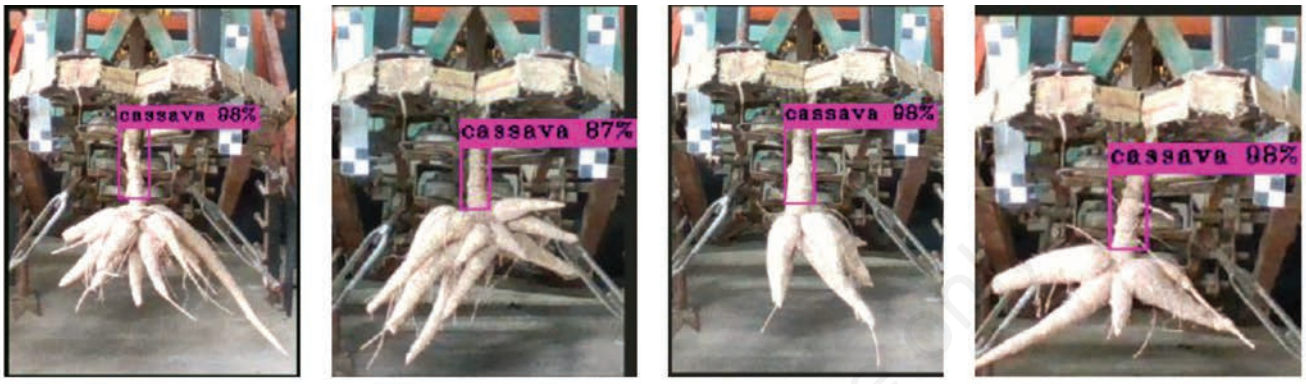

Figure 7. Localization of a cassava stalk by the object detection method.

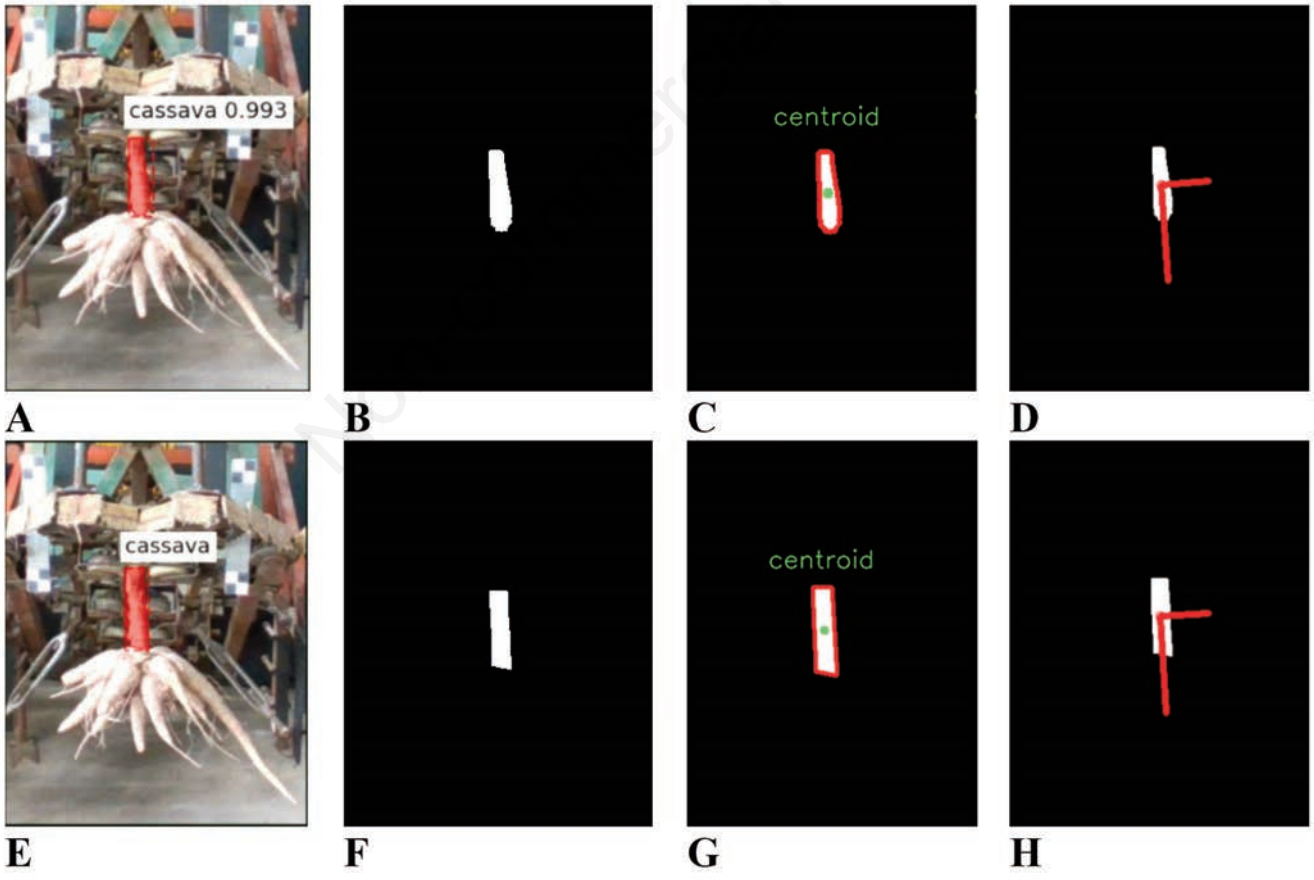

Figure 8. Predicted and target images: **A**) predicted segmentation image; **B**) predicted mask image; **C**) centroid and contour line of the predicted mask; **D**) predicted inclination; **E**) target segmentation image; **F**) target mask image; **G**) centroid and contour line of target mask; **H**) target inclination.

to less than two and tended to be stable, indicating that the training models were well run. The loss function essentially reached a state of convergence, as shown in Figure 10.

Evaluation of cassava stalk localisation performance

We compared the localisation performances of the three models, YOLO v4, Mask R-CNN (ResNet 101), and Mask R-CNN (ResNet 50). The main objectives were the grasping point and inclination detection. The bounding box (in YOLO v4) and the mask (in Mask R-CNN) were used to predict the grasping point and inclination. The standard of localisation performance was evaluated to select the best model.

The results from the 100 test images indicated that the average accuracy for all models was 1.00; since the target area of the image was very small, TN was large. The mean F1 score for Mask R-CNN with ResNet 101 was significantly higher than for YOLO v4 and Mask R-CNN with ResNet 50. The IoU mean for ResNet 101 was also significantly higher than the others. The predicted mask generated using Mask R-CNN with ResNet 101 was better than the ResNet 50 model because the number of convolutional layers in the training process of ResNet 101 was higher. The details of the segmentation performance evaluation are given in Table 1.

In agriculture, computer vision is normally used to detect target fruits, such as strawberries, oranges, apples, and mangos, from images containing many fruits. The results of such studies include the precision rate, recall rate, IoU, F1 score, and average precision (AP), as shown in Table 2. These values were calculated from a number of instances that gave both positive and negative results for fruit detection. Mask R-CNN has been used to predict the segmentation for images of strawberries: Yu *et al*. (2019) improved the fruit detection performance for a strawberry harvesting robot, with an MIoU of 89.85%, while Ge *et al*. (2019) developed a strategy for the localisation of strawberries and achieved an F1 score of 0.88. Ganesh *et al*. (2019) presented a deep-learning approach for detecting orange with Mask R-CNN from RGB and HSV images and obtained an F1 score of close to 0.89. Detection and segmentation of overlapping fruit based on an optimised Mask R-CNN application was carried out by Jia *et al*. (2020) in an apple harvesting robot, with a precision of 97.31% and a recall of 95.70%.

Our research represents the first step in developing an automatic cassava harvester; in this case, there is only one target object in each image. Therefore, the number of pixels was used for evaluation rather than the number of target objects. The F1 score and mean IoU for the YOLO v4 model were 0.73 and 0.58, respectively, whereas for the Mask R-CNN with ResNet 101, the F1 score and mean IoU were 0.81 and 0.70, respectively. For ResNet 50, the F1 score and mean IoU were 0.74 and 0.59, respectively. This means that the localisation performance of all of our models was good compared with prior schemes for other crops, and all three were acceptable for use in detecting cassava stalks.

Grasping point and inclination of predicted cassava stalks

The difference between the target and predicted coordinates (distance error) was calculated as the target position minus the predicted point. For the *x*-coordinate, a positive predicted value lies on the left side of the target point, whereas a negative predicted value lies on the right side. For the *y*-coordinate, a positive predicted value was higher than the target point, whereas a negative value was below the target point. For Mask R-CNN, most populations

Figure 9. A cassava stalk and the marker on the pulling machine.

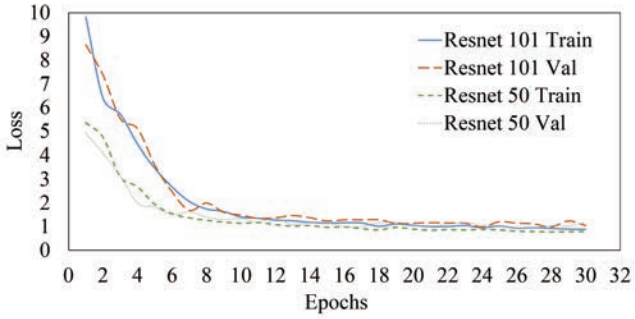

Figure 10. Change in the loss value over successive epochs.

Table 1. Evaluation of localisation performance.

Model		Mean accuracy	Mean F1 score	Mean IoU
YOLO v4		1.00[a]	0.73[b]	0.58[b]
Mask R-CNN	ResNet 101	1.00[a]	0.81[a]	0.70a
	ResNet 50	1.00[a]	0.74[b]	0.59[b]

[a,b]Values in the same column followed by the same letter are non-significantly different ($p<0.05$). IoU, intersection over union.

from the ResNet 101 and ResNet 50 models had a distance error in the *x*-coordinate within the range of 0-5 mm, and for the *y*-coordinate, these errors were in the same range. For the inclination result errors, most of the populations from ResNet 101 (87 samples) and ResNet 50 (84 samples) were within 0-5° degrees.

The correct distance and angle are necessary for gripping and alignment in the cassava root-cutting process. For 100 samples, the maximum distance error was 18.3 mm for the *x* coordinate and 57.5 mm for the *y* coordinate for both the ResNet 101 and ResNet 50 models. The maximum inclination errors for these models were 32.43° and 23.02°, respectively. For YOLO v4, the maximum distance errors for the *x*- and *y*-coordinates were 1.00 and 48.3 mm, respectively (all distances and inclinations were converted to absolute values). These values were considered the worst-case scenario in this study and may provide supporting data for gripping and alignment setting in the cassava root-cutting process.

The mean absolute error (MAE) in the distances of the *x*- and *y*-coordinates in all models was non-significantly different. This shows that all three models could predict the grasping point with the same error level. The values of the MAE for the inclination in ResNet 101 and ResNet 50 were non-significantly different, whereas the value for YOLO v4 was significantly worse, meaning that Mask R-CNN (Resnet 101 and 50) was more suitable for predicting the inclination than YOLO v4. Statistical data on the absolute errors in the distance and inclination are given in Table 3.

Prior studies that have presented results for the distance errors of targets have involved an apple and pear harvester, a sweet pepper harvester, and a strawberry picking machine, as shown in Table 4. The target object was predicted using the centroid and diameter for the apple and pear harvester, and the average displacement error was 10.6 mm (Font *et al*., 2014). For the sweet pepper harvester of Mooney and Johnson (2014), the average errors in the *x*- and *y*-directions were 1.3 and 25 mm, respectively. Although these two studies measured the distance or displacement error, they did not use a deep learning algorithm. However, Yu *et al*. (2019) used Mask R-CNN to detect the picking point for ripe strawberry fruits, with an average error of 1.20 mm.

In this study, the displacement errors were 12.5 and 9.8 mm for Mask R-CNN and YOLO v4, respectively. These errors were higher than for the strawberry picking machine, which used the same

Table 2. Localisation performance of crop detection methods in other studies using Mask R-CNN and YOLO.

Researchers	Model	Target crop	Results	
			F1	IoU
Yu *et al.* (2019)	Mask R-CNN	Strawberry	0.96	0.90
Ge *et al.*(2019)	Mask R-CNN	Strawberry	0.88	N/A
Ganesh *et al.* (2019)	Mask R-CNN	Orange	0.89	N/A
Jia *et al.* (2020)	Mask R-CNN	Apple	0.95	N/A
Tian *et al.* (2019)	YOLO	Apple	0.82	0.90
Koirala *et al.* (2019)	YOLO	Mango	0.95	N/A
Present study	Mask R-CNN	Cassava stalk	0.81	0.70
	YOLO v4	Cassava stalk	0.73	0.58

IoU, intersection over union.

Table 3. Localisation performance of crop detection methods in other studies using Mask R-CNN and YOLO.

Quantity	Model		N	Max	Min	Mean	SD
x axis (mm)	YOLO v4		100	10.00	0.00	3.50	2.70
	Mask R-CNN	ResNet 101	100	18.33	0.00	3.00	2.90
		ResNet 50	100	18.33	0.00	2.80	2.90
y axis (mm)	YOLO v4	100	48.30	0.00	8.30	9.30	
	Mask R-CNN	ResNet 101	100	57.50	0.40	11.8	13.3
		ResNet 50	100	57.50	0.00	10.9	13.5
Inclination (degrees)	YOLO v4	100	20.29	0.32	6.96	4.63	
	Mask R-CNN	ResNet 101	100	32.43	0.03	3.96	4.76
		ResNet 50	100	23.02	0.01	3.36	4.16

SD, standard error.

Table 4. Distance errors in studies of other crops.

Researchers	Algorithm	Crop(s)	Average error (mm)		
			x	*y*	Displacement
Font *et al.* (2014)	Hole filling	Apple and pear	N/A	N/A	10.60
Mooney and Johnson (2014)	Direct linear transformation	Sweet pepper	0.5	0.5	N/A
Yu *et al.* (2019)	Mask R-CNN	Strawberry	N/A	N/A	1.20
Present study	Mask R-CNN	Cassava stalk	3.00	11.80	12.50
	YOLO v4	Cassava stalk	3.50	8.30	9.80

model (Mask R-CNN) because the shape of the cassava stalk is very different from that of a strawberry. In addition, the length of a cassava stalk is also longer than its width. However, the displacement error was not the main objective of our study, as the parameters that will be used for the gripper controller are the coordinates on the *x*- and *y*-axes and the inclination. The distance errors in both the *x*- and *y*-directions, and the inclination are, therefore, the main results of this study.

An evaluation of the grasping point prediction performance is shown in Table 5. The values for R^2 between the predicted and target values of the *x*- and *y*-coordinates and the inclination for the ResNet 101 and ResNet 50 models show that the ResNet 50 model was slightly better than the ResNet 101 model, although the correlation was on the same level. The value of R^2 for the *x*-coordinate was 0.85 for both ResNet 101 and ResNet 50, whereas the values for the *y*-coordinate were 0.37 and 0.39, respectively, and the inclinations were 0.50 and 0.61, respectively, as shown in Figure 11A and C. For YOLO v4, the R^2 values for the *x*- and *y*-coordinates were 0.95 and 0.73, respectively, as illustrated in Figure 11B and D. Thus, the *x*-coordinate was satisfactorily predicted in all models, but the *y*-coordinate showed a poor correlation in Mask R-CNN and an acceptable correlation in the YOLO v4 model. For more details, in the 2D image, the dimension in the horizontal direction (parallel to the *x*-axis) represents the diameter of the cassava stalk in the actual situation. In contrast, the dimension in the vertical direction represents the distance from the hanging point (on the pulling machine) to the first cassava root (*i.e.*, the height of the cassava stalk). The diameters of the cassava stalks ranged from 20 to 70 mm, and the heights were between 100 and 150 mm. This means that the dimension of the cassava stalks in the vertical direction was more than twice as long as that in the horizontal direction, and hence the *y*-coordinate had a higher error than the *x*-coordinate (Figure 12). Figure 13A and B gives examples of poor predictions due to the irregular shape of the cassava stalk, where only the top area was detected (above the middle of the stalk). This resulted in poorly predicted segmentation (*i.e.*, the predicted value for *y* was far from the target value) hence a low accuracy for the predicted *y*-coordinate. Examples of good predictions are shown in Figure 13C and D, where the cassava stalk had a relatively symmetric shape (a rectangle, parallelogram, or trapezium, *i.e.*, not polygon-shaped). In the case of a poor prediction, the Mask R-CNN result can be improved by increasing the number of samples with irregular shapes in the training and validation sets.

Figure 11E shows that the R^2 values for the target and predicted inclinations of Mask R-CNN with ResNet 101 and ResNet 50 were 0.50 and 0.61, respectively, representing only a fair correlation because some specimens had irregular and unsymmetrical shapes. In addition, these specimens resulted in poorly predicted segmentation; thus, the predicted inclination had relatively low accuracy. For the YOLO v4 results, the inclination was not calculated but was defined based on the vertical axis, for which the inclination was set to 90° from the horizontal axis, as shown in Figure 11F. However, most populations (76 samples) of target inclinations were 85-100°, as illustrated in Figure 12. Therefore, these data should be carefully used and compared to other results.

An evaluation of the predictive ability of the model based on the RMSEP results showed that for the *x* coordinate, YOLO v4 was slightly worse than Mask R-CNN (ResNet 101 and ResNet 50), whereas for the *y* coordinate, Mask R-CNN with ResNet 101 gave the worst result, and YOLO v4 gave the worst result for inclination prediction, as shown in Table 5.

The predicted grasping point in *x* and *y* coordinates will be used as the grasping point for the gripper of a cassava root-cutting robot. Our previous study showed that the gripper could expand on the horizontal axis (*x*-coordinate) to 80 mm to grasp a stalk (Singhpoo *et al.*, 2019), whereas the diameters of the cassava stalks are 20 to 70 mm.

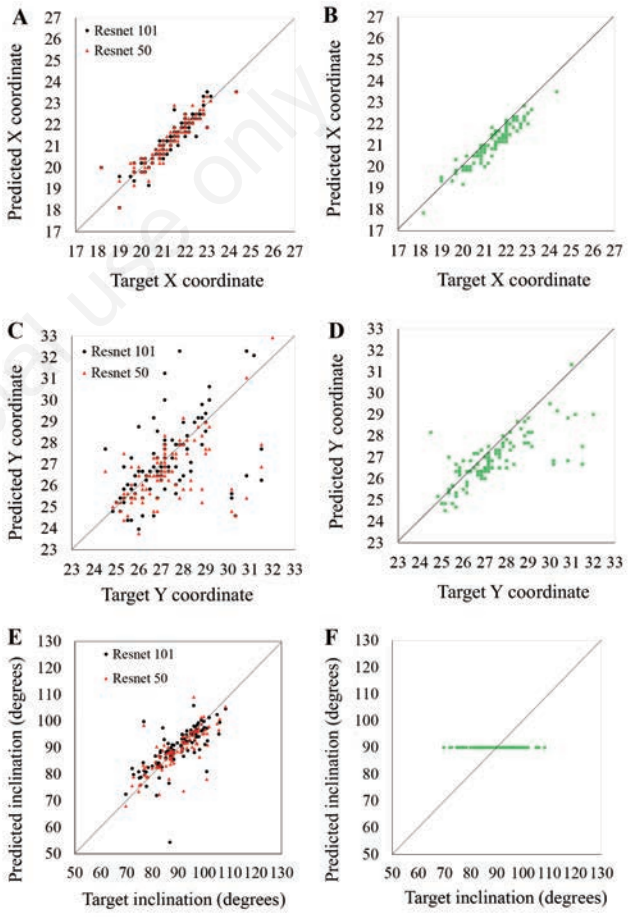

Figure 11. Correlation graph of predicted and target values: **A**) *x*-coordinate for Mask R-CNN; **B**) *x*-coordinate for YOLO v4; **C**) *y*-coordinate for Mask R-CNN; **D**) *y*-coordinate for YOLO v4; **E**) inclination for Mask R-CNN; **F**) inclination for YOLO v4.

Table 5. Performance in terms of grasping point and inclination prediction.

Model		*x*-coordinate		*y*-coordinate		Inclination	
		R^2	RMSEP	R^2	RMSEP	R^2	RMSEP
YOLO v4	0.89	4.40	0.53	12.5	N/A	8.36	
Mask R-CNN	ResNet 101	0.85	4.20	0.37	17.7	0.50	6.19
	ResNet 50	0.85	4.10	0.39	17.3	0.61	5.34

RMSEP, root mean square error of prediction.

The RMSEP for the *x*-coordinate from the YOLO v4 model was 4.40 mm (the highest value of all the models). This means the coordinate on the horizontal axis was of no concern, as the gripper could be expanded to more than 74.40 mm. Hence, the model in this study is acceptable for use with the gripper. On the vertical axis (*y*-coordinate), our previous study (Singhpoo, 2019) showed that the optimum dimension for the depth of the grasping arm was 50 mm, as shown in Figure 14. The heights of the cassava stalks are between 100 and 150 mm. Therefore, when the cassava stalk is grasped in the cutting process, the most appropriate point is the centre of the stalk area (75 mm
above the first root on the vertical axis). Since the depth of the grasping arm is 50 mm, its haft is 25 mm, and the highest position it can grasp is lower than 25 mm from the top position and higher than 25 mm from the lowest position. Consequently, the allowable distance errors for the gripper were ±25 and ±50 mm for cassava lengths of 100 and 150 mm, respectively. The optimal grasping position is shown in Figure 14. The RMSEP of the *y*-coordinate from the ResNet 101 model was 17.7 mm (the highest value of all models). This indicates that the models in this paper can be used with a gripper with the exact dimensions as those described above.

The inclination result is used as the cutting alignment for the robot. The value of RMSEP obtained from YOLO v4 was 8.36, and all specimen inclinations were 90° from the horizontal axis. This is not a precise operation and may not be appropriate for alignment with the cutter unit. In contrast, the inclinations predicted by Mask R-CNN with ResNet 101 and ResNet 50 were 6.19° and 5.34°, respectively. These smaller angles affect the cutting alignment only slightly; therefore, they could be used as input parameters for the cutter controller. Hence, the Mask R-CNN models can be used to predict the cutting alignment.

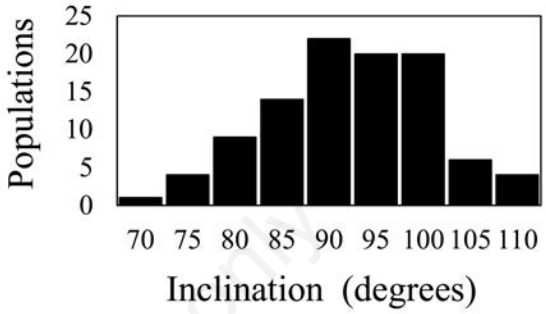

Figure 12. Target inclination distribution.

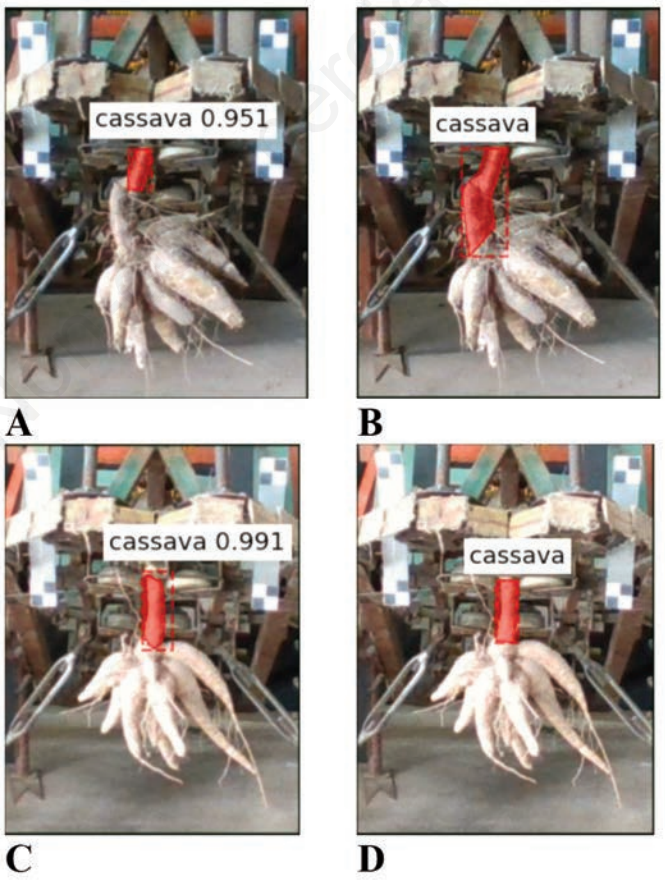

Figure 13. Segmentation results from Mask R-CNN: **A**) predicted segmentation of poor stalk; **B**) target segmentation of poor stalk; **C**) predicted segmentation of good stalk; **D**) target segmentation of good stalk.

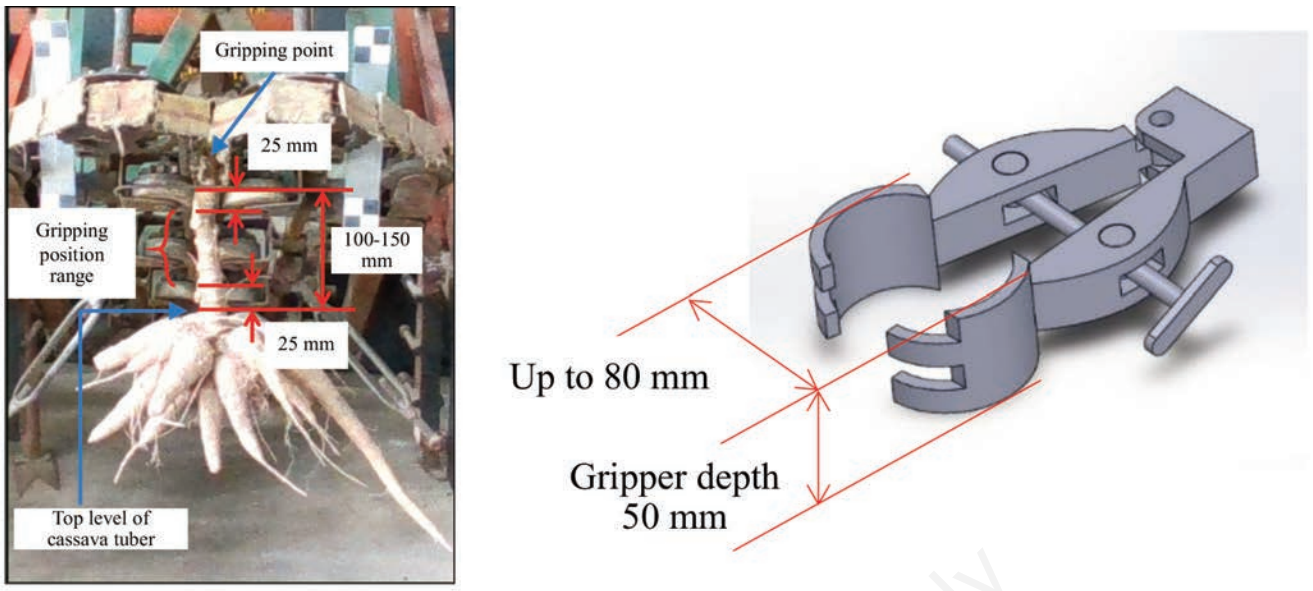

Figure 14. The optimal gripping point on a cassava stalk.

In general, the advantages of the Mask R-CNN model are the contour line and accurate shape that can be extracted from the image. In addition, an accurate grasping point and inclination were obtained from this model. However, the results show that all models' R^2 values for the *x*-coordinate were at the same level. The R^2 value for the *y*-coordinate in Mask R-CNN was worse than in the YOLO v4 model, but the inclination in YOLO v4 was only 90°.

In summary, although a precise grasping point can be calculated using either the Mask R-CNN or YOLO v4 models, only Mask R-CNN can predict the inclination well. Mask R-CNN is, therefore, the optimal model for localising the cassava stalk and predicting the grasping point and inclination.

Conclusions

This paper has examined the process of automatic cassava stalk detection for developing a cassava harvester. An object detection model (YOLO v4 model) and two segmentation models (Mask R-CNN with ResNet 101 and ResNet 50) were used to train the weights on images from the training set. Then, each weight was used to predict the cassava stalks in the test set. The prediction results from the object detection model consisted of a bounding box and its centre, whereas the results from the segmentation models were mask images. The centroid (*x*- and *y*-coordinates) and inclination of the cassava stalk were then calculated from the mask image. The centre of the bounding box and the centroid of the mask were compared with the mask centroid from the target image. The performance in terms of predicting the grasping point was evaluated using statistical indices such as R^2 and RMSEP, and the results indicated that the performance for prediction of the *x*-coordinate in all models [object detection (YOLO v4) and segmentation (ResNet 101 and ResNet 50 with Mask R-CNN)] was at the same level. Although object detection was the best approach for predicting the *y*-coordinate, the results for the inclination were worse than the alternative model. Overall, ResNet 101 was the best model: it yielded a high correlation for the *x*-coordinate, a low correlation for the *y*-coordinate, and an acceptable correlation for the inclination, with R^2=0.85, 0.37, and 0.50, respectively. The other index used was RMSEP, and the results indicated that the model could be used with a gripper in a cassava root-cutting robot. The values of RMSEP for the *x*-axis and the inclination were highest for the object detection method, at 4.40 mm and 8.36°, respectively. For the *y*-axis, ResNet 101 gave the highest value of 17.7 mm. The model is acceptable for use with the gripper or robotic arm described in a previous study. The model detected the optimal position on the stalk, the solid position. Fairly accurate alignment of the cutting process was also obtained with this model, meaning reliable cutting can be achieved using the proposed technique. In summary, the ResNet 101 model is the most suitable for the cassava root-cutting robot. These results will be helpful to researchers investigating cassava machines and can be applied to create a cassava root-cutting robot, reduce the harvesting time, increase the precision of work, and raise the quality of fresh cassava roots.

Nomenclature

Abbreviations

YOLO	You only look once
R-CNN	Region convolutional neural network
Faster R-CNN	Faster region convolutional neural network
Mask R-CNN	Mask region convolutional neural network
VIA	Visual geometry group image annotation
PNG	Portable network graphics
RGB-D	Red, green, blue, and depth
ResNet	Residual network
FCN	Fully convolutional network
RPN	Region proposal network
RoI	Region of interest
TP	True positive
TN	True negative
FP	False positive
FN	False negative

C	Centroid point
n	Number of pixels
PCA	Principal component analysis
IoU	Intersect over union
MAE	Mean absolute error
RMSEP	Root mean square error of prediction
N	Number of samples

Symbols

x_i	Pixel for calculating the centroid (pixels)
P_t	Pixels making up the cassava stalk in the target image (pixels)
P_p	Pixels making up the cassava stalk in the predicted image (pixels)
y_{ti}	Measured value of the target data
y_{pi}	Predicted values
$\overline{y}$	Average measured value of the target data
R^2	Coefficient of determination

References

Abdulla W. 2016. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. GitHub Repository. Available from: https://github.com/matterport/Mask_RCNN

Arsawang S., Chansrakoo W., Chamsing A., Sangphanta P., Chawkongchak S. 2015. Design and development of Cassava root plucking out machine. Agric. Sci. J. 47:463-6.

Bapat K. 2018. Find the center of a blob (Centroid) using OpenCV (C++/Python). LearnOpenCV; July 19, 2018. Available from: https://www.learnopencv.com/find-center-of-blob-centroid-using-opencv-cpp-python/

Blok P.M., Barth R., van den Berg W. 2016. Machine vision for a selective broccoli harvesting robot. IFAC-PapersOnLine 49:66-71. Available from: https://doi.org/10.1016/j.ifacol.2016.10.013

Boonsang, S. 2020a. CiRA CORE : community. KMITL. Available from: https://www.facebook.com/groups/cira.core.comm/

Boonsang S. 2020b. KSL cira core. KLS. Available from: https://sites.google.com/site/klsrobotcenter/kls-cira-core

Brownlee J. 2021. A gentle introduction to object recognition with deep learning. Available from: https://machinelearningmastery.com/object-recognition-with-deep-learning/

Chansiri, C., & Wongpichet, S. 2011. Research and development of the digging and gathering machine for cassava harvesting. Khon Kaen University, Khon Kaen, Thailand.

Dutta, A., & Zisserman, A. 2019. The VIA annotation software for images, audio and video. MM 2019 - Proceedings of the 27th ACM International Conference on Multimedia, 2276-2279. Available from: https://doi.org/10.1145/3343031.335053.

FAO. 2013. Cassava, a 21st century crop. In: Save and grow: cassava, a guide to sustainable production intensification. Available from: http://www.fao.org/ag/save-and-grow/cassava/en/1/index.html

Font D., Pallejà T., Tresanchez M., Runcan D., Moreno J., Martínez D., Teixidó M., Palacín J. 2014. A proposal for automatic fruit harvesting by combining a low cost stereovision camera and a robotic arm. Sensors (Switzerland) 14:11557-79.

Fu L., Majeed Y., Zhang X., Karkee M., Zhang Q. 2020. Faster R-CNN-based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting. Biosyst. Engine. 197:245-56.

Ganesh P., Volle K., Burks T.F., Mehta S.S. 2019. Deep orange: Mask R-CNN based orange detection and segmentation. IFAC-PapersOnLine 52:70-5.

Ge Y., Xiong Y., From P.J. 2019. Instance segmentation and localization of strawberries in farm conditions for automatic fruit harvesting. IFAC-PapersOnLine 52:294-9.

Girshick R. 2015. Fast R-CNN. pp 1440-1448 in Proceedings of the IEEE International Conference on Computer Vision 2015 Inter, 1. Available from: https://doi.org/10.1109/ICCV.2015.169

He K., Gkioxari G., Dollár P., Girshick R. 2020. Mask R-CNN. IEEE Trans. Pattern Anal. Machine Intell. 42:386-97.

Jia W., Tian Y., Luo R., Zhang Z., Lian J., Zheng Y. 2020. Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot. Comput. Electron. Agric. 172:105380.

Jordan J. 2018. Evaluating image segmentation. Available from: https://www.jeremyjordan.me/evaluating-image-segmentation-models/

Koirala A., Walsh K.B., Wang Z., McCarthy C. 2019. Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'MangoYOLO.' Precis. Agric. 20:1107-35.

KURDI. 2020. index @ www3.rdi.ku.ac.th. KU. Available from: http://www3.rdi.ku.ac.th/?p=58386

Langkapin J., Kalsirisilp R., Tantrabandit M. 2012. Design and fabrication of cassava root picking machine. Agric. Sci. J. 30:1-11.

Ling X., Zhao Y., Gong L., Liu C., Wang T. 2019. Dual-arm cooperation and implementing for robotic harvesting tomato using binocular vision. Robot. Auton. Syst. 114:134-43.

Majeed Y., Karkee M., Zhang Q., Fu L., Whiting M.D. 2019. A study on the detection of visible parts of cordons using deep learning networks for automated green shoot thinning in vineyards. IFAC-PapersOnLine 52:82-6.

Manthamkan, V., Rattanasrimetha, S., & Suriwong, M. 2011. Development of cassava root lifted up by pulling stump harvester type. Karsetsart Extension Journal, 56(2), 52–60. Available from: https://kukrdb.lib.ku.ac.th/journal/ETO/search_detail/result/43687.

Mao S., Li Y., Ma Y., Zhang B., Zhou J., Kai W. 2020. Automatic cucumber recognition algorithm for harvesting robots in the natural environment using deep learning and multi-feature fusion. Comput. Electron. Agric. 170:105254.

Maraphum K., Saengprachatanarug K., Wongpichet S., Phuphaphud A., Posom J. 2020. In-field measurement of starch content of cassava tubers using handheld vis-near infrared spectroscopy implemented for breeding programmes. Comput. Electron. Agric. 175:105607.

Mauntumkarn W. 2010. Cassava harvesting machine. Available from: http://www.rdi.ku.ac.th/kasetresearch53/group06/wichar/index_04html

Mooney J.G., Johnson E.N. 2014. Performance evaluation of a harvesting robot for sweet pepper. J. Field Robot. 33:1-17.

Mordvintsev A., Abid K. 2013. opencv python. Available from: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_contours/py_contours_begin/py_contours_begin.html

OAE. 2021. Agricultural economic information by commodity. Office Agric. Econ. Available from: http://www.oae.go.th/view/1/ปัจจัยการผลิต/TH-TH

Redmon J., Divvala S., Girshick R., Farhadi A. 2016. You only look once: unified, real-time object detection. pp. 779-788 in Proceedings of the IEEE Computer Society Conference on

Computer Vision and Pattern Recognition. Available from: https://doi.org/10.1109/CVPR.2016.91

Redmon J., Farhadi A. 2018. YOLO v.3. Tech Report, 1–6. Available from: https://pjreddie.com/media/files/papers/YOLOv3.pdf

Ren S., He K., Girshick R., Sun J. 2017. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Machine Intell. 39:1137-49.

Sangphanta P., Chansrakoo W., Arsawang S., Chamsing A. 2011. Research and development of cassava harvesting machine. p. 12 in Proceedings of the 12th Thai Society of Agricultural Engineering Annual Academic Meeting.

Saxena S. 2021. Image augmentation techniques for training deep learning models. Analytics Vidhya. Available from: https://www.analyticsvidhya.com/blog/2021/03/image-augmentation-techniques-for-training-deep-learning-models/

Singhpoo T. 2019. Factors affecting cassava root cutting by using cylinder saw mechanism. Khon Kaen Univesity, Khon Kaen, Thailand.

Singhpoo T., Wongpichet S., Saengprachatanarug K., Posom J., Watyotha C., Yangyuen S. 2019. A study of stalk shape for designing the operational mechanism of gripping equipment for cassava tuber cut preparation process. IOP Conf. Ser. Earth Environ. Sci. 301:1-6.

Suvanapa K., Wongpichet S. 2014. Feasibility study of cassava rhizome cutting by using square tube blade. pp. 335-342 in The 16th Thai Society of Agricultural Engineering Annual Academic Meeting.

Tian Y., Yang G., Wang Z., Wang H., Li E., Liang Z. 2019. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Comput. Electron. Agricult. 157:417-26.

Vatakit K., Somphong C., Junyusen P., Arjharn W. 2014. Development of cassava harvester for cutting cassava tuber from rhizome. Agric. Sci. J. 45:353-6.

Williams H.A.M., Jones M.H., Nejati M., Seabright M.J., Bell J., Penhall N.D., Barnett J.J., Duke M.D., Scarfe A.J., Ahn H.S., Lim J.Y., MacDonald B.A. 2019. Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms. Biosyst. Engine. 181:140-56.

Yu Y., Zhang K., Yang L., Zhang D. 2019. Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN. Comput. Electron. Agric. 163:104846.