Jurnal Kejuruteraan 33(4) 2021: 1113-1121
https://doi.org/10.17576/jkukm-2021-33(4)-34

A Combinatorial RGB and Depth Images CNN-based Model for Oil Palm Fruit Bunch Detection and Heatmap Localisation for a Visual SLAM System

Chua Zheng Siong[a], Mohd Faisal Ibrahim[a*], Aqilah Baseri Huddin[a], Mohd Hairi Mohd Zaman[a] & Fazida Hanim Hashim[a]

[a]*Department of Electrical, Electronic and Systems Engineering,*
Faculty of Engineering & Built Environment, Universiti Kebangsaan Malaysia, Malaysia

**Corresponding author: faisal.ibrahim@ukm.edu.my*

Received 12 March 2021, Received in revised form 11 June 2021
Accepted 12 July 2021, Available online 30 November 2021

ABSTRACT

The harvesting job of cutting and collecting fruit bunches in oil palm plantations remains the most labour-intensive job in the oil palm processing cycle. The introduction of an autonomous vehicle to assist workers in the harvesting job promises better productivity. Such a driverless vehicle requires a software module known as simultaneous localisation and mapping (SLAM) to guide the vehicle to navigate autonomously. This work proposes a visual SLAM system with a distinctive capability of detecting and localising oil palm loose fresh fruit bunches (FFB) on the ground using intelligent image processing. This vehicle is equipped with a depth camera capable of capturing RGB images and depth images concurrently. Two VGG16-based convolutional neural network (CNN) models are trained using the acquired RGB and depth images dataset of loose FFBs on the ground. The output from the combinatorial FFB detection model is then fed into a visual SLAM system called RTAB-Map. By combining the FFB detection model and the visual SLAM system, the vehicle can plan for autonomous navigation safely, perform bunch pick-up tasks, and avoid collision with fruit bunches on the ground. The experiment results show that the proposed CNN model can detect and localise loose FFBs with significant accuracy in various lighting conditions.

Keywords: Oil palm fruit bunch detection; deep learning model; convolutional neural network; visual SLAM; depth camera object detection.

INTRODUCTION

Palm oil is a sustainable crop and a major contributor to global vegetable oil demand. It can yield usable oil at least six times more efficiently than other major crops like soybean, sunflower, and canola (Kojima et al. 2016; Murphy 2014).

The harvesting job of oil palm fresh fruit bunches (FFB) can be considered a challenging and high-risk job. This is due to the nature of the labour that requires the cut and collection of heavy bunches with approximate weights between 10 to 24 kg per bunch (Harun and Noor 2002). Due to regular lifting, poor body postures, and repetitive tasks in the daily work, the burden on the worker would cause an ergonomic hazard. This ergonomic hazard harms the labourer's musculoskeletal system and brings a long-term effect on their health condition (Nawi et al. 2016).

To improve the process of oil palm harvesting, mechanised tools are required to improve productivity (Aljawadi et al. 2018, Khalid et al. 2021). One of the modern tools is an autonomous plantation vehicle (Pedersen et al. 2016). This driverless vehicle can be utilised in various ways, including carrying cutting tools, collecting fruit bunches, cleaning up plantation areas, and fertilising trees. The vehicle requires a software module known as simultaneous localisation and mapping (SLAM) to guide the vehicle to navigate autonomously. Apart from the SLAM system, an important sub-module module that can detect loose FFBs on the ground is required. The detection module has at least two functions, 1) to provide detection output to the vehicle's collision avoidance software to prevent collision with the bunches, 2) to detect the location of the bunches so that mechanised actions such as collecting the bunches can be done.

This work proposes the application of a depth camera,

sometimes known as RGB-D camera, installed on such vehicle and convolutional neural network (CNN) models to detect loose FFBs on the ground. A depth camera is a type of camera that can capture RGB format images and depth images concurrently CNN models are artificial intelligence-based models that can be trained with a machine learning approach known as deep learning.

Subsequently, the fruit bunch detection module will be combined with a visual SLAM system called RTAB-Map to produce an occupancy grid map that can be used by the vehicle to plan its autonomous navigation (Labbé and Michaud 2016; Silva et al. 2018). The visual SLAM system refers to the process of mapping an unknown environment around the sensor and simultaneously determining the location of the vehicle and the orientation of the sensor in the map (Das 2018).

The contribution of this work is two-fold. First, unlike any conventional object detection, the proposed loose FFB detection model was developed based on RGB-D images which are the combination of RGB images and depth images. The proposed model utilises the combination of visual features provided by RGB images and the geometry feature provided by depth images to ensure accurate detection of loose FFBs. The fusion of both features has shown better image recognition in various outdoor light levels, especially in agriculture applications (Cruz et al. 2012, Gai et al. 2020).

Second, a deep learning algorithm with the fusion of convolutional neural network (CNN) models implementing transfer learning of VGG16 network (Simonyan and Zisserman 2014) for an RGB image model and a depth image model were utilised (Zhao et al. 2019). The main advantage of the CNN model is the automation of detecting important features while learning without human intervention.

Different harvesting technologies have been reported in the literature (Sowat et al. 2018, Yusoff et al. 2019, Khalid et al. 2021), including telescopic mechanical arms, climbing robots, suction mechanism collectors, and roller picker robots. The proposed loose FFB detection is possible to be integrated into such machines to provide advancement in the assisted semi-automatic or fully automatic control mechanism.

This paper is organised as follows. The next section discusses the methods used to develop the oil palm fruit bunch detection module combined with RTAB-Map based visual SLAM. The following section presents the results and discusses the findings, including the detection models' training phase based on real bunch images and the testing of the combined detection and SLAM modules based on a virtual simulation setup. Finally, the main findings from this work are summarised in the Conclusion section.

METHODOLOGY

This section is divided into three sub-sections. The first sub-section describes the development of the oil palm fruit bunch detection model. Second, the localisation function of fruit bunch on an image is presented. The third sub-section focuses on the combinatorial of the fruit bunch detection model with the RTAB-Map visual SLAM system.

OIL PALM FRUIT BUNCH DETECTION MODEL

In this work, the model of oil palm fruit bunch detection is proposed by using CNN architecture based on a custom network in which the structure of the network is inspired from the VGG-16 architecture involving convolution, pooling, flatten, dropout, and dense layers. The customisation is done on the full-connected layers for classification purposes. To build the model, an open-source machine learning platform known as TensorFlow and Keras was used. TensorFlow is a back-end framework that is responsible for performing deep learning calculations. Keras is a front-end library written in Python that provides a user-friendly implementation of deep learning. Figure 1 shows the flowchart of the processes taken to build CNN models for loose FFBs detection.

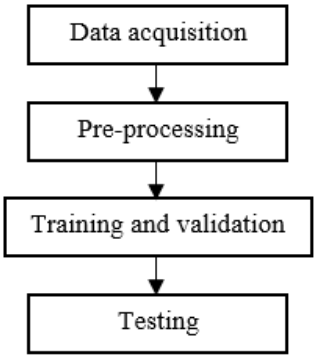

FIGURE 1. CNN models development processes

First, RGB image and depth image samples of a real oil palm loose FFBs were taken by using a depth camera called Kinect Xbox 360. The dimension of both image types is 640 x 480 pixels. The image format is in Joint Photography Experts Group (JPG). A total of 394 RGB images and 314 depth images had been captured as raw image data, taken in bright and dark ambient lights. For RGB images, 222 images contain loose FFBs while the other 172 images contain no fruit bunches. Meanwhile, there are 172 loose FFBs images and 142 non-loose FFBs images for the depth images category.

After the data acquisition step, an image pre-processing step is required. The acquired data were split

into training, validation, and testing groups. For the RGB images, the dataset was divided into 245, 105, and 44 images for training, validation, and testing group, respectively. Meanwhile, for the depth images, the dataset was separated into 189, 81, and 44 images for the training, validation, and testing group, respectively. Note that, in the testing group, each RGB image has its corresponding depth image. Figure 2 shows samples of pair-images of loose FFBs in different levels of light.

Another image pre-processing step is image resizing. The 640 x 480 pixels (width x height) original images were changed to 224 x 224 pixels to fit with the VGG16 network structure. The RGB images consist of three (3) colour channels, while the depth images consist of only one (1) colour channel. Again, the depth images were transformed into three (3) channel data to fit with the VGG16 network structure. Furthermore, an image generator was used to generate more training data by image zooming, rotating, and brightness leveling processes.

Once the dataset is ready, a training and validation step is performed. This core process determines the configuration of CNN models by deep learning using the pre-processed training dataset. The CNN models used in this work have the same feature extraction layers as in the VGG16 network structure with thirteen (13) convolutional layers and five (5) pooling layers. For the classification layers, the proposed model uses one (1) flatten layer, one (1) dropout layer, and (2) dense layer. Figure 3 and Table 1 show the proposed CNN model structure and the detailed configuration of each layer used in both the RGB model and the depth model, respectively.

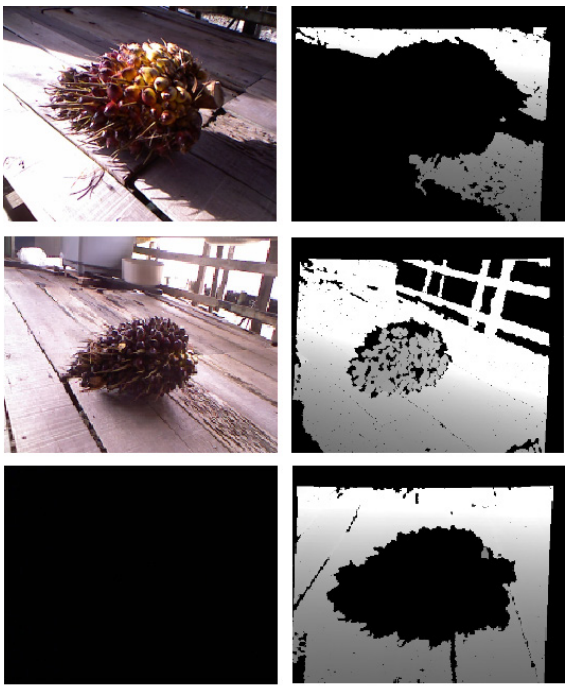

FIGURE 2. Loose FFBs in different levels of light

The chosen optimiser for the training of the model is "Adam", a gradient-based optimisation technique. The selected loss function and the performance metric parameters are "binary cross entropy" and "accuracy", respectively.

The sigmoid activation function is chosen to model the binary classifier. Binary classification is the task of classifying the elements of a given set into two groups based on the classification rule. In this work, there are two classification groups, loose FFBs and non-FFBs. The justification to using the binary classification is to optimise features learning capability whereby the inputs from two different classes can maximise inter-class difference and minimise intra-class variance.

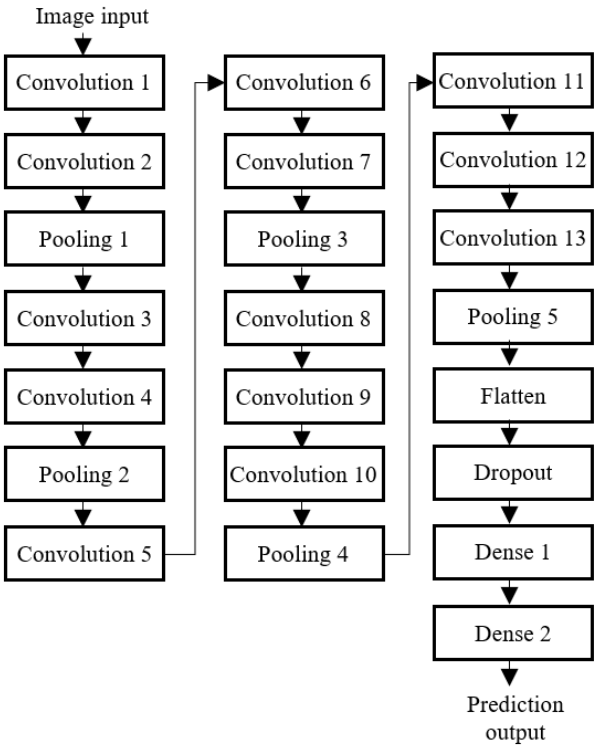

FIGURE 3. The proposed CNN model structure based on VGG16 network for the RGB image detection model and depth image detection model

With the above configuration, the final configuration of the trained CNN models is chosen from models with the lowest validation loss. The abovementioned training-and-validation step is applicable for both the RGB image detection model and the depth image detection model.

After observing acceptable training loss and accuracy performance indices, verification is done using testing data in the final testing step. The testing process involves the calculation of true positive rate (TPR) and false positive rate (FPR) from a confusion matrix that describes the performance of a classifier (Piegorsch 2020). The values of TPR and FPR are calculated by using (1) and (2).

$$TPR = \frac{TP}{TP+FN} \quad (1)$$

$$FPR = 1 - \frac{TN}{TN+FP} \quad (2)$$

where TP is the number of true positive cases, TN is the number of true negative cases, FP is the number of false positive cases and FN is the number of false negative cases.

The performance of the RGB image CNN model and the depth image CNN model was tested in various bunch positions and light conditions, to evaluate the importance of using the combination of RGB image and depth image models.

TABLE 1. The detailed configuration of each layer in the proposed CNN model. Parameters of convolution layers are stated in the form: (number of filters, filter size, stride, padding, and activation function). Parameters of pooling layers are stated in the form: (pooling type, pooling size, and stride).

Layer	Parameter	Output
Convolution 1 & 2	64, 3x3, 1, Same, ReLU	Feature map 224x224x64
Pooling 1	Max-pooling, 2x2, 2	Feature map 112x112x64
Convolution 3 & 4	128, 3x3, 1, Same, ReLU	Feature map 112x112x128
Pooling 2	Max-pooling, 2x2, 2	Feature map 56x56x128
Convolution 5, 6, 7	256, 3x3, 1, Same, ReLU	Feature map 56x56x256
Pooling 3	Max-pooling, 2x2, 2	Feature map 28x28x256
Convolution 8, 9, 10	512, 3x3, 1, Same, ReLU	Feature map 28x28x512
Pooling 4	Max-pooling, 2x2, 2	Feature map 14x14x512
Convolution 11, 12, 13	512, 3x3, 1, Same, ReLU	Feature map 14x14x512
Pooling 5	Max-pooling, 2x2, 2	Feature map 7x7x512
Flatten	-	Vector 26,088
Dropout	Drop rate: 0.3	Vector 26,088
Dense 1	Hidden nodes: 512 Activation: ReLU	Vector 512
Dense 2	Output node: 1 Activation: Sigmoid	Binary class 0: no FFB 1: has FFB

Note that both the RGB and depth image CNN models must be run in parallel to predict the presence of a fruit bunch. The detection output of both models was combined using the 'OR' gate concept. If one of the models detected a loose FFB, then the system will report that a fruit bunch exists.

OIL PALM FRUIT BUNCH LOCALISATION

The oil palm fruit bunch localisation in an image is accomplished based on the heatmap concept. Heatmap is a class activation map visualisation technique that shows the importance of each pixel for an input image relative to its output class in a two-dimensional grid diagram. The higher the value of the pixel, the more important the pixel is, relative to its output class. It is generated from the feature map of the final convolution layer in the CNN models. Figure 4 shows an example of the heatmap of the oil palm bunch RGB and depth images.

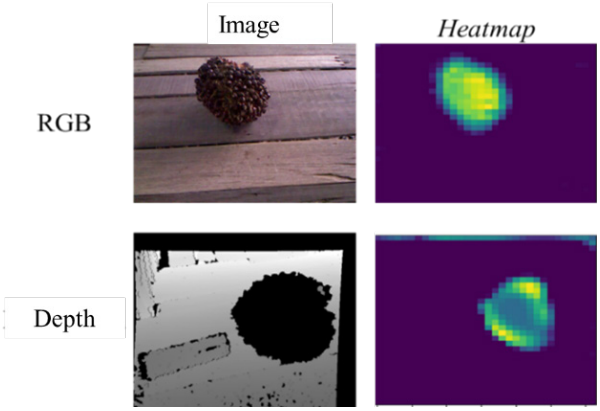

FIGURE 4. Examples of fruit bunch RGB and depth images with their corresponding heatmaps

Based on the generated heatmap, the distance between the camera and the oil palm bunch can be calculated. As the data originated from depth images, every pixel value contains the distance information between the object and the camera. Therefore, after the bunch is detected, the pixel in the heatmap which contains the highest value is selected as the centre location of the oil palm fruit bunch. However, the pixel position needs to be converted to an approximate position in the depth image by using (3) and (4) since the size of the heatmap is only 32 x 32 pixels.

$$x = \frac{position\ x\ in\ the\ heatmap}{32} \times 640 \quad (1)$$

$$y = \frac{position\ y\ in\ the\ heatmap}{32} \times 480 \quad (2)$$

After that, the pixel position can be translated into the position of the oil palm fruit bunch in an occupancy grid map of SLAM by using (5), (6) and (7) in relation to the location of the camera.

$$\sigma = (\frac{640}{2} - x) \times \frac{\sigma_f}{640} \quad (1)$$

$$o_x = c_x + D\cos(\theta + \sigma) \quad (2)$$

$$o_y = c_y + D\sin(\theta + \sigma) \quad (3)$$

where σf is the horizontal field of view (FoV) angle of the camera, which is 57° (green line), and σ is the angle between the camera and the fruit bunch. ox and oy are the coordinates of the oil palm bunch on the x-axis and y-axis, respectively. cx and cy are the location of the camera. The location of the camera is obtained from the odometry data in RTAB-Map. D is the distance between the oil palm bunch and the camera. θ is the orientation of the vehicle. The illustration of the calculation is shown in Figure 5.

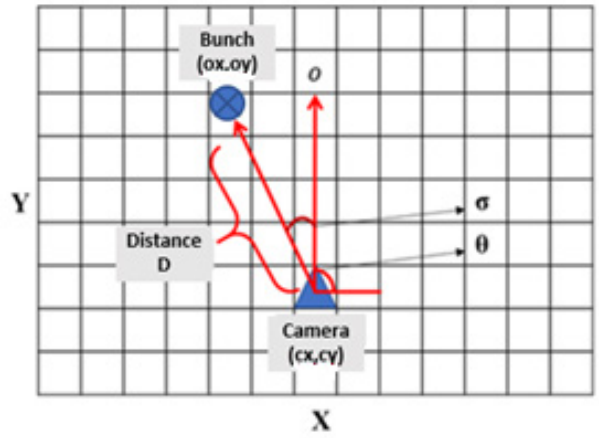

FIGURE 5. The illustration of oil palm fruit bunch location relative to the camera and vehicle position in an occupancy grid map of a SLAM

SIMULATION PLATFORM FOR MODELS VALIDATION

Gazebo simulator (Koenig and Howard 2004) was used to build a virtual world to test the developed oil palm fruit bunch detection and localisation functions. A virtual world with four compartments, as in Figure 6, was created. Each compartment has a different level of detection and localisation challenges in terms of the number of FFBs as well as the background of the environment.

To imitate the oil palm fruit bunch, a three-dimensional oil palm fruit bunch model was built by using Blender software. The simulator was then integrated with an open-source Robot Operating System (ROS), a robotic framework with a collection of tools, libraries, and conventions for various robotic functions and tasks. The RTAB-Map package in ROS was integrated with the simulated virtual world to provide SLAM module.

Turtlebot 2 model was selected as the virtual autonomous vehicle in the simulation for running all the developed programs. Kinect Xbox 360 camera is a part of the Turtlebot 2 system that provided the RGB-D images.

In the virtual world, Turtlebot 2 was wandered from one room to another. Simultaneously, the RTAB-Map node collected RGB-D images from the robot's camera, performed the three-dimensional mapping task, and updated a two-dimensional occupancy grid map.

RESULTS AND DISCUSSION

This section presents the results obtained from the CNN models development processes, specifically on the training and validation step as well as the testing step. After that, the results on applying the fruit bunch detection and localisation functions in the virtual world are discussed.

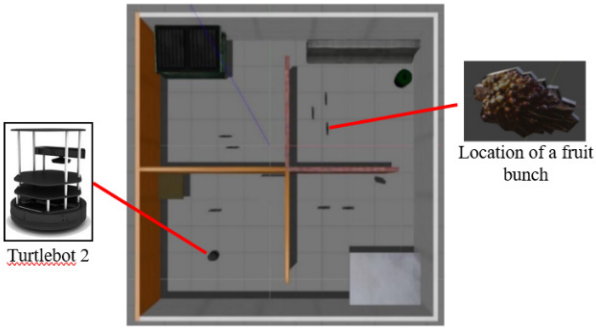

FIGURE 6. The virtual world with Turtlebot 2 in Gazebo

PERFORMANCE OF THE TRAINING AND VALIDATION STEP FOR THE FRUIT BUNCH DETECTION MODEL

A reliable CNN model should deliver high accuracy and low loss detection performance. Accuracy describes the percentage of the test data that is correctly classified while loss is the sum of differences between the predicted probabilities of the test data with 0 (non-oil palm bunches) or 1 (oil palm bunches). In this work, loss plays a more important role compared to accuracy. This is because accuracy is indistinguishable so it cannot be used for backpropagation of learning algorithms, while loss can be distinguished so it can act as a good proxy for accuracy. Therefore, a low loss means that the model has high accuracy and a good training process. This is the reason the Checkpoint Model function is used to store the model with the lowest validation loss.

Figure 7 shows the performance graph of loss and accuracy during the training and validation steps for the RGB image CNN model. The training was run for 100 epochs. Based on the performance graph, the model has the lowest validation loss at 0.1029.

On the other hand, Figure 8 depicts the performance graph of loss and accuracy during the training and validation step for the depth image CNN model. Using the same procedure of loss performance for the selection of the model, it is found that at epoch 72 the model gives the lowest loss at 0.0378. Thus, this model is chosen as the depth image CNN detection model.

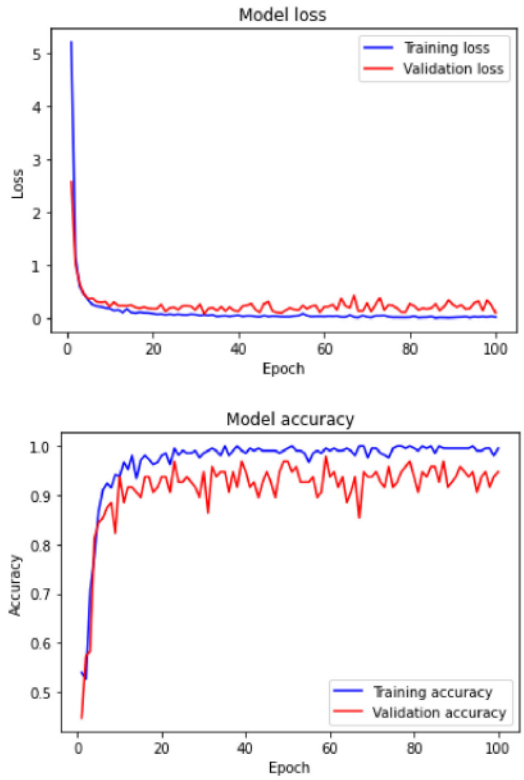

FIGURE 7. The graphs of loss and accuracy performance during the training-and-validation step for the RGB image CNN model.

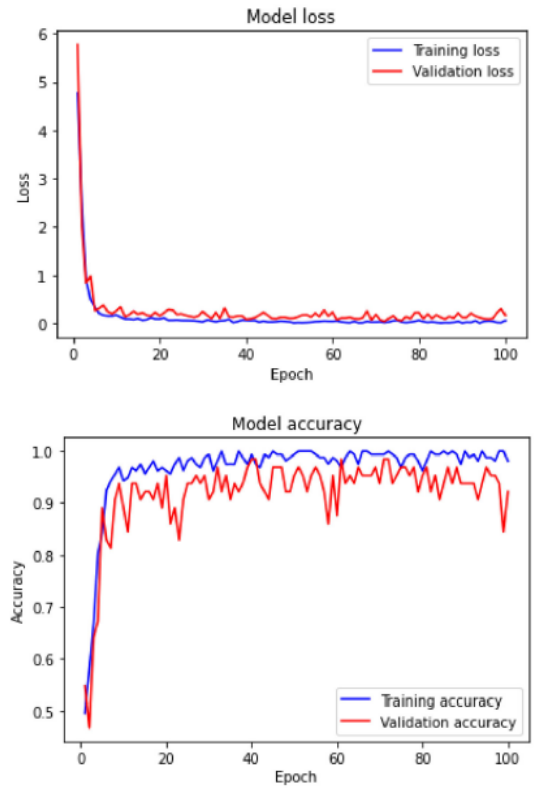

FIGURE 8. The graphs of loss and accuracy performance during the training-and-validation step for the depth image CNN model.

PERFORMANCE OF THE TESTING STEP FOR THE FRUIT BUNCH DETECTION MODEL

In this sub-section, the performance of the RGB image model and the depth image model are reported. Firstly, the confusion matrix is used to measure the performance of the trained models with the testing datasets. 44 images were used that contain 22 images with fruit bunch and another 22 images without fruit bunch. The RGB image and the depth image models were tested under various lighting conditions. The threshold value of the models' prediction output is set at 0.5. A prediction value that is more than 0.5 indicates the detection of a fruit bunch and vice-versa. Table 2 and Table 3 show the confusion matrix after applying the testing data on the RGB image model and the depth image model, respectively.

TABLE 2. Confusion matrix of the RGB image model

N = 44	Prediction: No fruit bunch	Prediction: Has fruit bunch
Actual: No fruit bunch	22	0
Actual: Has fruit bunch	5	17

TABLE 3. Confusion matrix of the depth image model

N = 44	Prediction: No fruit bunch	Prediction: Has fruit bunch
Actual: No fruit bunch	21	1
Actual: Has fruit bunch	3	19

After that, the TPR values and the FPR values of both models were calculated using (1) and (2). Table 3 shows the TPR and FPR values of the RGB model and the depth image model, respectively.

TABLE 4. TPR and FPR of the RGB image and depth image models

Model	TPR	FPR
RGB model	0.773	0.000
Depth model	0.864	0.045

Based on the calculation, the RGB image model and the depth image model had the TPR value at 0.773 and 0.864, respectively. This result indicates that both models can detect the presence of a fruit bunch well. However, the RGB image model fails to detect the bunch in a complete dark condition where 5 out of 22 images with bunch were.

For the depth image model, 3 out of 22 images cannot detect the fruit bunch. All these three images have the same image features in which the proximity measurement between the bunch and the background cannot be distinguished. This condition may occur due to the random error of depth measurement and low resolution of the camera.

For the non-fruit bunch images, the models return very low FPR values near zero. The results indicate that both models output low probability value for non-fruit bunch images on the testing data.

PERFORMANCE OF THE COMBINATORIAL RGB AND DEPTH IMAGE DETECTION MODELS

Light condition is one of the major factors affecting the accuracy of the oil palm fruit bunch detection. Figure 9 and Figure 10 show the example of the prediction of the RGB image model and the depth image model in bright and dark conditions. The image on the left side was taken in a bright condition while the image on the right side was taken in a dark condition. Note that the RGB image model can detect the presence of a fruit bunch in a bright condition and cannot detect the same bunch in a total dark condition. The performance of the depth image model is vice-versa on both light conditions.

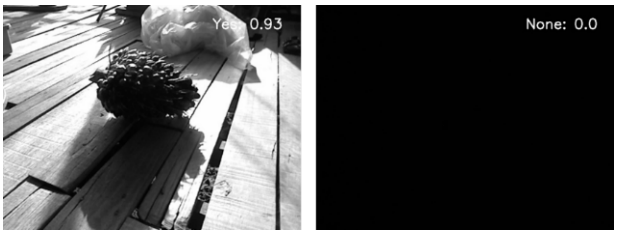

FIGURE 9. Detection output of the RGB image model for two samples: a bright image (left) and a dark image (right). Both images contain a fruit bunch.

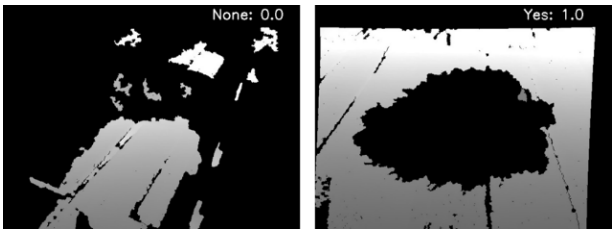

FIGURE 10. Detection output of the depth image model for two samples: a bright image (left) and a dark image (right). Both images contain a fruit bunch.

Based on the results, the performance of the depth image model is lower than the RGB image model in a bright condition. The area irradiates to the direct sunlight in the depth image becomes black as sunlight contains infrared light. Accordingly, the Kinect camera is not designed to operate under strong lighting conditions (Alenyà et al. 2014). Thus, when the oil palm bunch is exposed to sunlight, the features of the oil palm bunch are lost and cannot be detected correctly. Although the obtained RGB image is also influenced by sunlight, the effect is not strong enough to affect its performance. The RGB image model can still detect the oil palm bunch accurately.

However, the depth image model shows better performance in a dark condition compared to the RGB image model. A depth image taken in a dark condition was able to show the shape and the feature of the oil palm fruit bunch. Thus, it can detect the presence of oil palm fruit bunch correctly because the depth information obtained by the infrared light is not affected by the light condition.

In order to get better performance, the RGB image model and the depth image model are integrated to form an RGB-D image model. The prediction of the RGB-D image model comes from the prediction of the RGB image model combined with depth image models through the concept of 'OR' gate. This means that if one of the models detects an object as an oil palm fruit bunch, then the object will be classified as an oil palm fruit bunch.

In other words, the RGB-D image model is less affected by light intensity, hence it is best to use it for this application. Based on the results, the RGB image model prediction is more successful than the depth image model in an environment with sufficient light intensity. In contrast, the depth image model works well in low light environments.

To validate the final model, the testing data were applied again. The output from the combinatorial model was obtained for each testing image. Table 5 tabulates the confusion matrix of the model, indicating the prediction performance. It is clear that the combinatorial model perfectly predicts the correct class for each image, either containing a fruit bunch or not. Thus, the highest possible TPR value of 1.000 and the lowest possible FPR value of 0.000 were achieved

TABLE 5. Confusion matrix of the combinatorial RGB and depth images model

N = 44	Prediction: No FFB	Prediction: Has FFB
Actual: No FFB	22	0
Actual: Has FFB	0	22
TPR	1.000	
FPR	0.000	

Figure 11 shows the three-dimensional map (from the top view) produced by the RTAB-map package with fruit bunch detection and localisation functions for the virtual world in Figure 6. The map was obtained after Turtlebot 2 wandered the four compartments completely. The locations of the oil palm fruit bunch were marked by the yellow dots on the map.

After that, the detection and localisation performance was examined in the corresponding two-dimensional occupancy grid map. Figure 12 shows the occupancy grid map produced with the oil palm fruit bunch detection and localisation functions, respectively. In the occupancy grid map, the black-coloured area means obstacles, the light gray-coloured area means free area, and the dark gray-coloured area means area not discovered yet by the robot's camera.

From the figure, the locations of the fruit bunch landmarks can be detected and localised in the grid map. The locations of oil palm fruit bunch were marked clearly as obstacles indicated by black-coloured areas. The number of fruit bunch was also correctly counted by the system. In conclusion, by using the RTAB-Map with the combination of the fruit bunch detection and localisation functions, the oil palm bunches can be detected, located, and marked as obstacles in the occupancy grid map for preventing the autonomous vehicle from passing through them and causing damage to the fruit bunches.

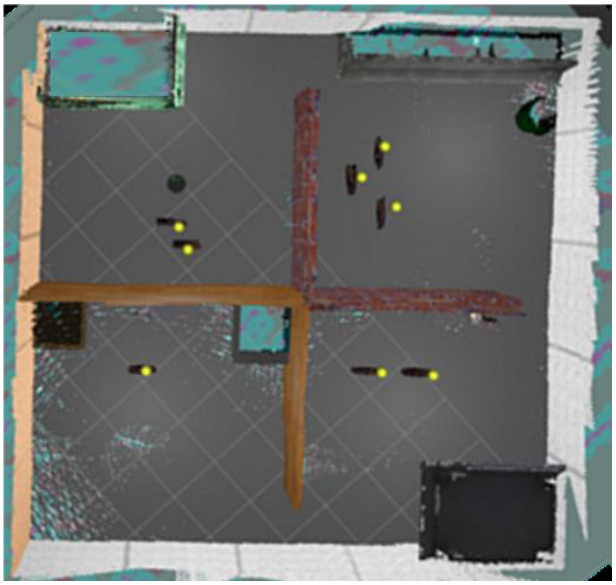

FIGURE 11. Three-dimensional map produced by RTAB-Map with the fruit bunch detection and localisation functions.

FIGURE 12. An occupancy grid map with the fruit bunch detection and localisation functions.

CONCLUSION

Oil palm fruit bunch detection and localisation functions have been developed based on RGB images and depth images. Deep learning of convolutional neural networks is implemented to train, validate, and test the functionality of the two developed models, which are the RGB image model and the depth image model. The developed models have high detection accuracy when combined to form a combinatorial RGB-D image detection model. The model is also robust to the variant of ambient light condition. Furthermore, the fruit bunch detection and localisation functions are integrated into a visual SLAM known as RTAB-Map to produce a three-dimensional map and a two-dimensional occupancy map with the capability to detect and localise fruit bunches. The produced map can be beneficial to the navigation system of an autonomous vehicle to avoid the vehicle from passing through the detected fruit bunches. Further works may be required to extend the current limitation for adoption in various plantation surroundings. It is suggested that the fruit bunch detection models to be trained with various types of oil palm fruit bunches in terms of different sizes, shapes, backgrounds, and light conditions to widen the application of the system according to the types of oil palm fruit bunches. In addition, a multi-bunch detection algorithm can be considered to reflect the real-world implementation.

ACKNOWLEDGEMENT

The authors would like to thank Universiti Kebangsaan Malaysia for the financial support under the grant GUP-2019-018, the Department of Electrical, Electronic and Systems Engineering UKM and UKM-Yayasan Sime Darby Chair for the technical supports. Authors would also like to thank the committee of FYP JKEES for the guidance and project management of this work.

DECLARATION OF COMPETING INTEREST

None

REFERENCES

Aljawadi, R.A., Ahmad, D., Mat-Nawi, N. & Saufi, M. 2018. Mechanized harvesting of oil palm fresh fruit bunches: A review. *In Proc. Conference: Capacity Building in Agriculture, Forestry and Plantation,* Bangi.

Alenyà, G., Foix, S. & Torras, C. 2014. Using ToF and RGBD cameras for 3D robot perception and manipulation in human environments. *Intelligent Service Robotics* 7(4):211–220.

Cruz, L., Lucio, D. & Velho, L. 2012. Kinect and RGBD images: Challenges and applications. *In Proc. 25th Conference on Graphics, Patterns and Images*, pp. 36–49.

Das, S. 2018. Simultaneous Localization and Mapping (SLAM) using RTAB-MAP. *International Journal of Scientific and Engineering Research* 9(8).

Gai, J., Tang, L. & Steward, B.L. 2020. Automated crop plant detection based on the fusion of color and depth images for robotic weed control. *Journal of Field Robotics* 37(1): 35-52.

Harun, M.H. & Noor, M.R.M. 2002. Fruit set and oil palm bunch components oil palm bunch components. *Journal of Oil Palm Research* 14(2): 24-33.

Khalid, M.R., Shuib, A.R. & Kamarudin, N. 2021. Mechanising oil palm loose fruits collection – A review. *Journal of Oil Palm Research* 33(1):1-11.

Koenig, N. & Howard, A. 2004. Design and use paradigms for gazebo, an open-source multi-robot simulator. *In Proc. IEEE/RSj International Conference on Intelligent Robots and Systems* 2149-2154.

Kojima, Y., Parcell, J. & Cain, J. 2016. A Global Demand Analysis of Vegetable Oils for Food and Industrial Use: A Cross-Country Panel Data Analysis with Spatial Econometrics. *In Proc. Agricultural and Applied Economics Association Annual Meeting,* Massachusetts.

Labbé, M. & Michaud, F. 2019. RTAB-Map as an open-source lidar and visual slam library for large-scale and long-term online operation. *Journal of Field Robotics* 36(2): 416–446.

Murphy, D.J. 2014. The future of oil palm as a major global crop: Opportunities and challenges. *Journal of Oil Palm Research* 26(1): 1-24.

Nawi, N.S.M., Deros, B.M., Rahman, M.N.A., Sukadarin, E.H. & Nordin, N. 2016. Malaysian oil palm workers are in pain: Hazards identification and ergonomics related problems. *Malaysian Journal of Public Health Medicine* 16(1): 50–57.

Pedersen, S.M., Fountas, S. & Blackmore, B.S. 2008. Agricultural Robots - Applications and Economic Perspectives. *Service Robot Applications*, Y. Takahashi, Ed. Croatia.

Piegorsch, W.W. 2020. Confusion Matrix. in *Wiley StatsRef:* Statistics Reference Online.

Silva, D., Cooray, B.P., Chinthaka, J.I., Kumara, P.P. & Sooriyaaracchi, S.J. 2018. Comparative Analysis of Octomap and RTABMap for Multi-robot Disaster Site Mapping. *In Proc. 18th International Conference on Advances in ICT for Emerging Regions,* Colombo.

Simonyan, K. & Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *In Proc. 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA.*

Sowat, S.N., Wan-Ismail, W.I., Mahadi, M.R., Bejo, S.K. & Mohd-Kassim, M.S. 2018. Trend in the development of oil palm fruit harvesting technologies in Malaysia. *Jurnal Teknologi* 80(2):83-91.

Yusoff, M.Z.M., Zamri, A., Abd-Kadir, M.Z.A., Wan Hassan, W.Z. & Azis, N. 2019. Loose fruit collector machine in Malaysia: A review. *International Journal of Engineering Technology and Sciences (IJETS)* 6(2): 2462-1269.

Zhao, Z.Q., Zheng, P., Xu, S.T. & Wu, X. 2019. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems* 30(11):3212 - 3232.