International Journal of Robotics and Automation, Vol. 38, No. 10, 2023
An Open Access Paper

USING YUMI ROBOT AND RGB-D CAMERA WITH YOLOV5 FOR PICK-AND-PLACE APPLICATION

Dumrongsak Kijdech* and Supachai Vongbunyong*

Abstract

Nowadays, in many industries, robots and cameras are used together to detect certain objects and perform specific tasks. However, misdetection can be occurred due to uncertainty of lighting condition, background, and environment. Using a dual arm 7-DOF collaborative robot and RGB-D camera with YOLOv5 in pick-and-place application is proposed in this research to resolve the aforementioned problems. The images are collected and labelled in preparation of the dataset. The dataset is trained with the machine learning algorithm, YOLOv5. It became weight for real-time detection. When RGB images from the camera are sent to YOLOv5, data in regard to position x–y and colour of the bottle is extracted from the depth and the colour images. The experiments were done to assess the performance of YOLO and the grasping capability of the robot.

Key Words

Yumi, convolution neural networks, YOLO, RGB-D camera, artificial intelligent

1. Introduction

These days, the demand for robots and automation has increased dramatically in many industries due to labour shortage and high wage rate. Robots and automation can resolve these issues. However, general image processing package with industrial robots has limitation when dealing with complex lighting and background. As a result, errors will prone while those condition is occurred.

Currently, a vision system with AI has gained more interest due to the improvement of the performance of computation speed of the graphic card on the computer. Previously, image processing is one of the most popular techniques for object classification and localisation. Recently, convolution neural network (CNN) has been widely used for object detection and classification. However, CNN is primarily divided into two types such as You Only Look Once (YOLO) and mask region-based CNN (R-CNN).

A number of researches were conducted in regard to the integration between AI vision system and industrial robots. However, integration between the AI vision system and collaborative robot that can deal with uncertainties in environment and objects has not been investigated. This research proposes using a dual-arm 7- degree-of-freedom (DOF) collaborative robot, "YuMi", and RGB-D camera with YOLOv5 for pick-and-place application. The working principle of this research is as follows: the camera will send images to YOLOv5. The position of the object is received and sent to the YuMi robot. The robot will pick the object and place to a specified position. To resolve the problem regarding the uncertainty of light and environment, the object detection and classification by using artificial intelligence are used. The ability of an AI to resist the uncertainty comes from training with a lot of data, images with various conditions of light and environment. YOLOv5 is faster in comparison to other YOLOs.

The rest of this paper is organised as follows: Section 2 discusses the literature review about robots with vision system and vision system with CNN and YOLOv5. Section 3 provides a methodology of the system setup: components, system overview and camera calculation. Section 4 presents the methodology of object detection with YOLOv5. Section 5 presents the experiment and Section 6 is the conclusion.

2. Literature Review

This section reviews the literature in regard to the application of the robot with vision system, vision system with CNN, and the principle of YOLOv5.

2.1 Review Robot with Vision System

A lot of research works implement robot arms with vision system to perform specific tasks, *e.g.*, [1]–[7]. Zakhama *et al.* [1] used the SCARA robot with image processing.

* Institute of Field Robotics (FIBO), King Mongkut's University of Technology Thonburi, Bangkok, Thailand; e-mail: dumrongsak.kijdech@mail.kmutt.ac.th; Supachai.von@kmutt.ac.th
Corresponding author: Supachai Vongbunyong

Recommended by Xin Xu
(DOI: 10.2316/J.2023.206-0927)

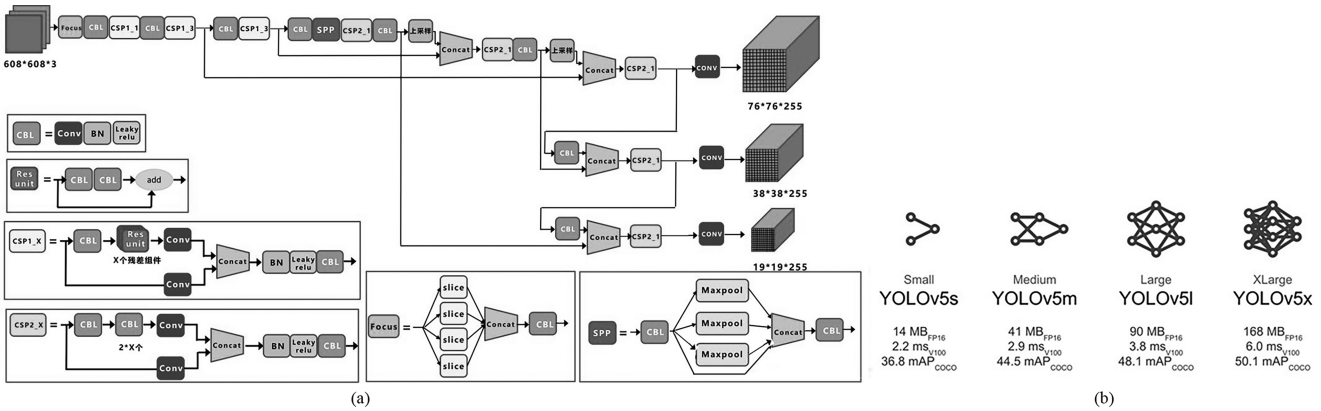


Figure 1. The images are labelled in an alphabetical: (a) structure of YOLOv5 [18] and (b) YOLOv5 different model sizes [20].

RGB image is converted to HSV image and then the histogram is analysed to find the objects and their types of defects. Kirschner *et al.* [2] used a dual-arm collaborative robot, YuMi, with an RGB-D camera to detect the object by using "PatMax Pattern" algorithm to find the edges of the object. Liang *et al.* [3] proposed using dual quaternion-based kinematic control to control YuMi robot. YuMi is a dual-arm 7-DOF collaborative robot. Controlling 7-DOF robots generally encounters problems in regard to joint limitation which can be solved by using various methods, *e.g.*, numerical approach [8]. Due to the safety issue, collaborative robots are suitable for applications that work alongside with human operators. Wu and Hong [4] and Yang *et al.* [5] used robots with YOLO on ROS with Python for detecting objects. Yang *et al.* [5] used the UR5 robot arm and intel RealSense D435 camera with YOLOv3 to detect object and grasp. YOLO was great for recognising and localising the objects when training with augmented images with various object orientation. Opaspilai *et al.* [6] used YOLOv3 with a SCARA robot to depalletisation of pharmaceutical product. Lelachaicharoeanpan and Vongbunyong [7] used YOLOv5 with UR5 robot to classify surgical devices. From the review and assessment, YOLOv5 is the most appropriate option for this research as it can deal with complicated environment.

2.2 Review Vision System with CNN

A number of object detection techniques are considered in this research. A number of research works [9]–[16] studied about performance of each object detection technique. References [9]–[13] and [16]–[18] studied various versions of YOLO from YOLOv1 to YOLOv5. Fang *et al.* [9] needed real-time objection, so that tinier-YOLO was suitable according to its small model size but a decrease in precision is a major drawback. Du [10] compared each object detection based on CNN family and YOLO. The training time and detection framerate are improved as the version is more updated. Sang *et al.* [11] and Zhang *et al.* [12] used YOLOv2 for vehicle detection and Chinese traffic sign with the fastest detection speed at 0.038 and 0.017 s per image, respectively. Tekin *et al.* [15] presented a development of YOLOv2 from 2D prediction to 6D prediction in real time. Zhao and Li [16] improved YOLOv3 for achieving better mean average precision (mAP) of object detection. However, in addition to CNN and YOLO, there are other artificial intelligence vision algorithms developed for particular tasks, *e.g.,* VGG-16 [19].

Although both techniques are based on CNN, YOLO is faster than Faster R-CNN [13], [14] because YOLO detects the objects from feature maps while Faster R-CNN is region based. The performance of YOLO keeps improving as newer versions were proposed, YOLOv4 and YOLOv5 [17], [18].

2.3 YOLOv5 Principle

YOLOv5 is based on CNN [structure shown in Fig. 1(a)] consisting of convolution (Conv) layers, batch normalisation (BN) layers, and Leaky rectified linear unit (ReLU) in many layers. YOLOv5 neural networks provide different model with different configurations and each parameter [see Fig. 1(b)], where FP16 stands for the half floating-point precision, V100 is an inference time in milliseconds on the Nvidia V100 GPU, and mAP based on the original COCO dataset. YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x are the structure of neural network model. Each structure of neural network with higher complexity use longer time for training process but offer higher accuracy for detection process, respectively. The mAP values for COCO dataset is from 36.8 to 50.1 and the inference time on the Nvidia V100 GPU is from 2.2 to 6.0 ms. In this research, YOLOv5l neural network model is used for the experiment.

3. Methodology: System Setup

Methodology is described in two sections. This section describes the system setup in regard to hardware and software, including each component, configuration of system, and camera calculation.

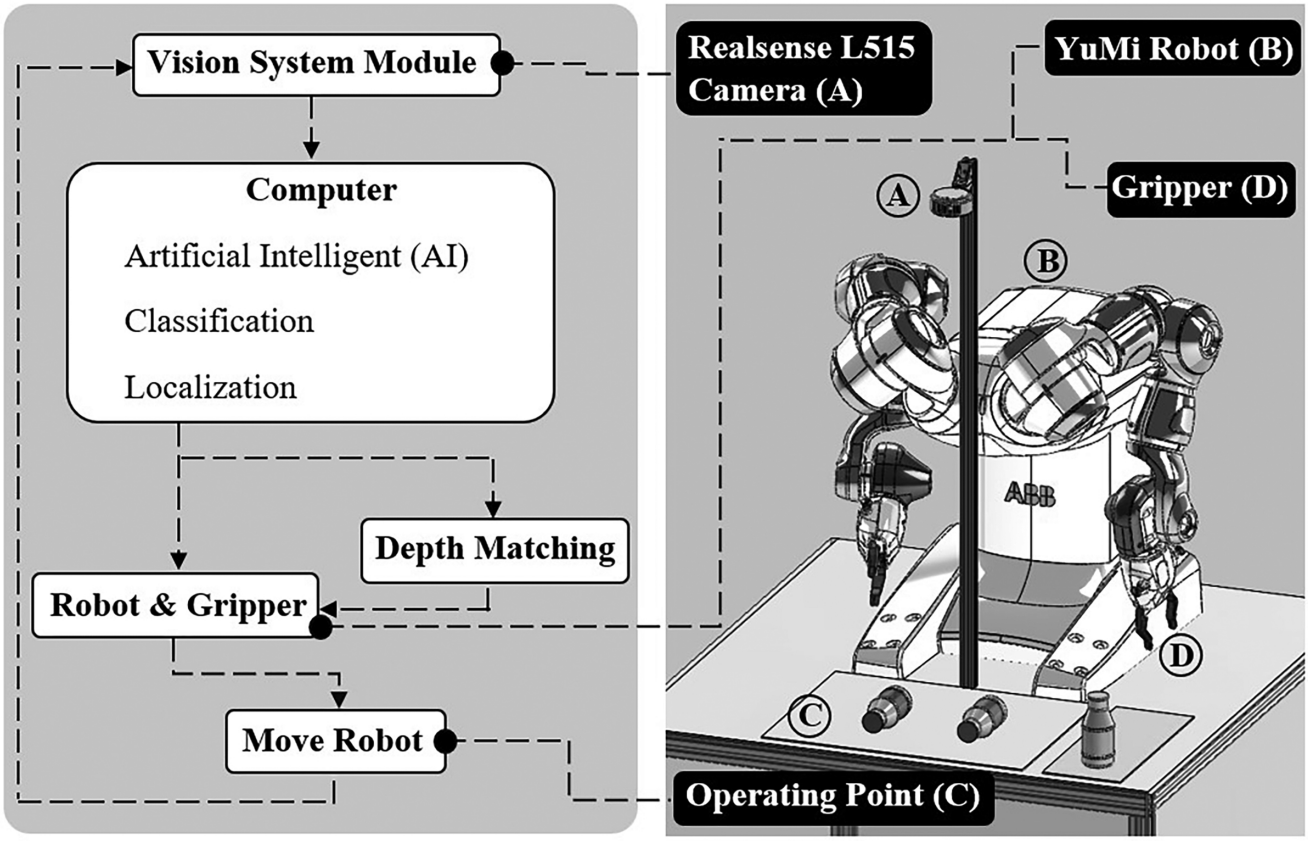


Figure 2. System overview: (a) diagram of this system and (b) configuration of this system.

3.1 Description of Each Component

The configuration of pick-and-place system is shown in Fig. 2(b). In this research, ABB IRB-14000 or "YuMi" is controlled with IRC5 controller. This robot is designed for working side-by-side with human operators. YuMi is suitable about assembling small parts with a payload of 500 g each arm. The workspace of this system is 560 mm × 320 mm on XY-plane in front of the robot as shown in Fig. 2(b). The friction gripper is designed specifically for grasping bottles and other objects as shown in Fig. 2(b). Realsense L515 is used as an RGB-D camera equipped above the robot [see in Fig. 2(b)]. This camera is selected according to its high resolution and its compact size. For the depth camera, minimum sensing distance is 250 mm with 5–14 mm depth accuracy. The depth image resolution is 1024 × 768 pixel and RGB image resolution is 1920 × 1080 pixel at 30 fps.

3.2 System Overview

Diagram of pick-and-place system of this research is shown in Fig. 2(a) and system diagram is shown in Fig. 3. Real-time RGB and depth images are sent from the camera to the computer to perform object detection and classification. Location of object (x, y, z, $q1$, $q2$, $q3$, $q4$) is sent from computer to the robot by a LAN cable (TCP/IP protocol). The process consists of three sections as follows.

(a) In *object detection process*, a colour image from the camera is processed by YOLOv5 to obtain the class and the location of object. The location of the centre of the object on the XY-plane is obtained from the colour image while the position on Z-axis is obtained from the depth image at the associated XY position.

(b) For *grasp planning algorithm*, the position XYZ of the object in camera coordinate obtained from the previous process is transformed to the robot coordinate. The Euler angles (row, pitch, and yaw) are converted to quaternion $q1$, $q2$, $q3$, and $q4$. The output of the object location will be managed by trajectory planning system afterwards. For the trajectory planning, the robot will grasp the object from above along the z direction. Planning of pick and place is as follows: starts from the standby posture, moves to the pre-grasp position, grasps the object, moves back to pre-grasp position, moves back to standby posture, moves to pre-place, place position, move back to pre-place position and then moves back to standby posture. Many via points are created in order to avoid the singularity of the robot arm.

(c) *Trajectory planning* is done on the computer with Python program. The set of command, such as x, y, z, q1, q2, q3, q4, and state of gripper, is sent to the robot via TCP/IP protocol with socket messaging.

Transformation matrix is used to transform the location of the object between coordinates. The reference coordinate of YuMi robot {B}, lenses centre of the camera {L}, product {P}, and fixture base {F} are shown in

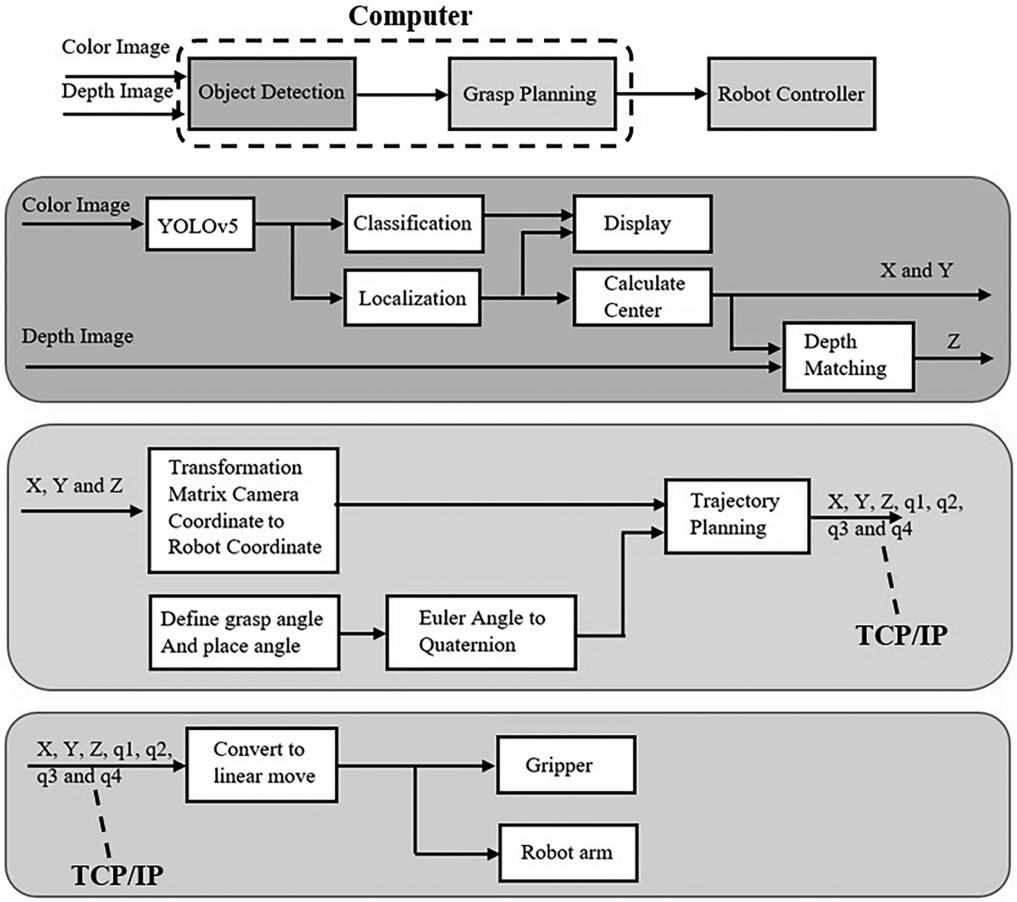


Figure 3. System overview of this research.

Fig. 4. The object position relative to {B} can be obtained from the transformation matrices in (1), where each term is shown in (2). P^L_{Object} is obtained by substituting (2) in (1), resulting in (3) which $L^B_{F,x}$, $L^B_{F,y}$, and $L^B_{F,z}$ are 232 mm, 0 mm, and 780 mm, respectively.

$$P^B_{\text{object}} = T^B_F T^F_L P^L_{\text{object}} \tag{1}$$

α_x and α_y are the scale factor to calibrate physical and captured image. α_x and α_y are mm per pixel.

$$T^B_F = \begin{bmatrix} -1 & 0 & 0 & L^B_{F,x} \\ 0 & -1 & 0 & L^B_{F,y} \\ 0 & 0 & 1 & L^B_{F,z} \\ 0 & 0 & 0 & 1 \end{bmatrix}; T^F_L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_{\text{LF}} \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$\text{and } P^L_{\text{object}} = \begin{bmatrix} P^L_{x,\text{object}} \\ P^L_{y,\text{object}} \\ P^L_{z,\text{object}} \end{bmatrix} \tag{2}$$

$$P^B_{\text{object}} = \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} = \begin{bmatrix} \left[\frac{1}{\alpha_x f}\left(L_{\text{LF}} - z_F(c,r)\right)\right](c - X_0) + L^B_{F,x} \\ \left[\frac{1}{\alpha_y f}\left(L_{\text{LF}} - z_F(c,r)\right)\right](-r + Y_0) + L^B_{F,y} \\ z_F(c,r) + L^B_{F,z} \end{bmatrix} \tag{3}$$

3.3 Camera Calculation

According to Fig. 2, the camera is installed above the robot and points downward to the working area. The full frame of colour image from the camera is shown in Fig. 4. However, original image is very large. Only the ROI is concerned to reduce computation time and disturbance. The origin point of the image coordinate refer to the line of sight of the camera. It is at $x = 160$ mm and $y = 300$ mm when measured according to the ROI frame. The resolution is reduced to 700 × 370 pixel and both images can be aligned with position offset.

4. Methodology: Object Detection with YOLOv5

This section presents the methodology regarding the object detection with YOLOv5, especially for dataset training. However, this research used only custom dataset for training, evaluation, and real-time detection.

4.1 System Overview

To detect the object with YOLOv5, the images from real-time camera are sent to convolution in Fig. 1. These images are flattened from $n \times m$ matrix to $nm \times 1$. After that the data is sent to neural networks model (YOLOv5l).

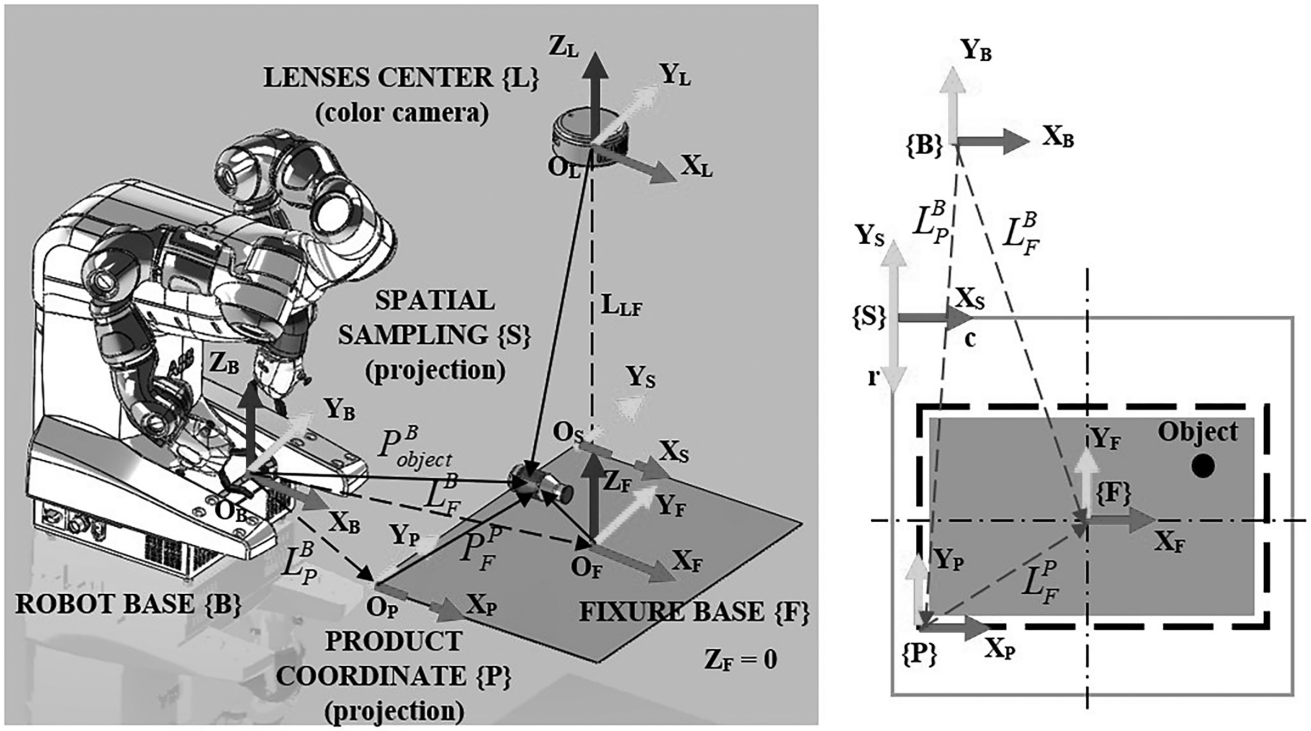


Figure 4. Coordinate of YuMi robot and frame coordinate.

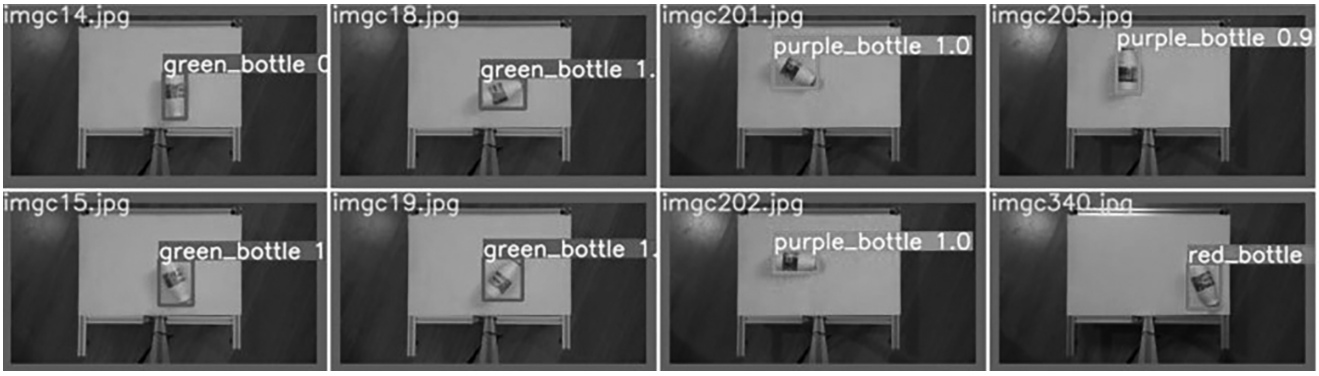


Figure 5. The dataset with label examples.

Eventually, the output is bounding box and class of the image.

Many versions of YOLO were tested for preliminary evaluation. YOLOv5s and YOLOv5m are found to have too small neural network structure for this application. Three versions of YOLOv5 – *i.e.*, YOLOv5m, UOLOv5l, and YOLOv5x – were compared. As a result, YOLOv5m was faster than YOLOv5l for training and detection but mAP is lower. Meanwhile, the training and detecting duration was almost double for YOLOv5x in comparison to YOLOv5l. In conclusion, YOLOv5l is selected.

4.2 Training Set

The RGB image is used for training while the depth image is used for localisation. The RGB images of the object with various postures and backgrounds are captured and resized. Examples of dataset with label for training is shown in Fig. 5. Also, the modified dataset is prepared to be used for an augmentation which is helpful to train the model that is capable of handling more complexity.

The object used in this research is a milk bottle with a colour label. They are chosen to be testing samples because the size, shape, and weight are suitable for the robot's capacity (500 g maximum payload). The object is around 200 g, and the shape is symmetrical. The colour label is different for each bottle, which is advantage for increasing the mAP as the training set is more variety. In total, 564 custom images were trained.

For the training process, the backbone of YOLOv5 is pytorch operated on Python programme. The data is trained by using GPU from Google collaborator and Tesla T4, 15109.75MB graphic card. This model is 499 layers and the number of YOLOv5l parameters is 46,642,120. The

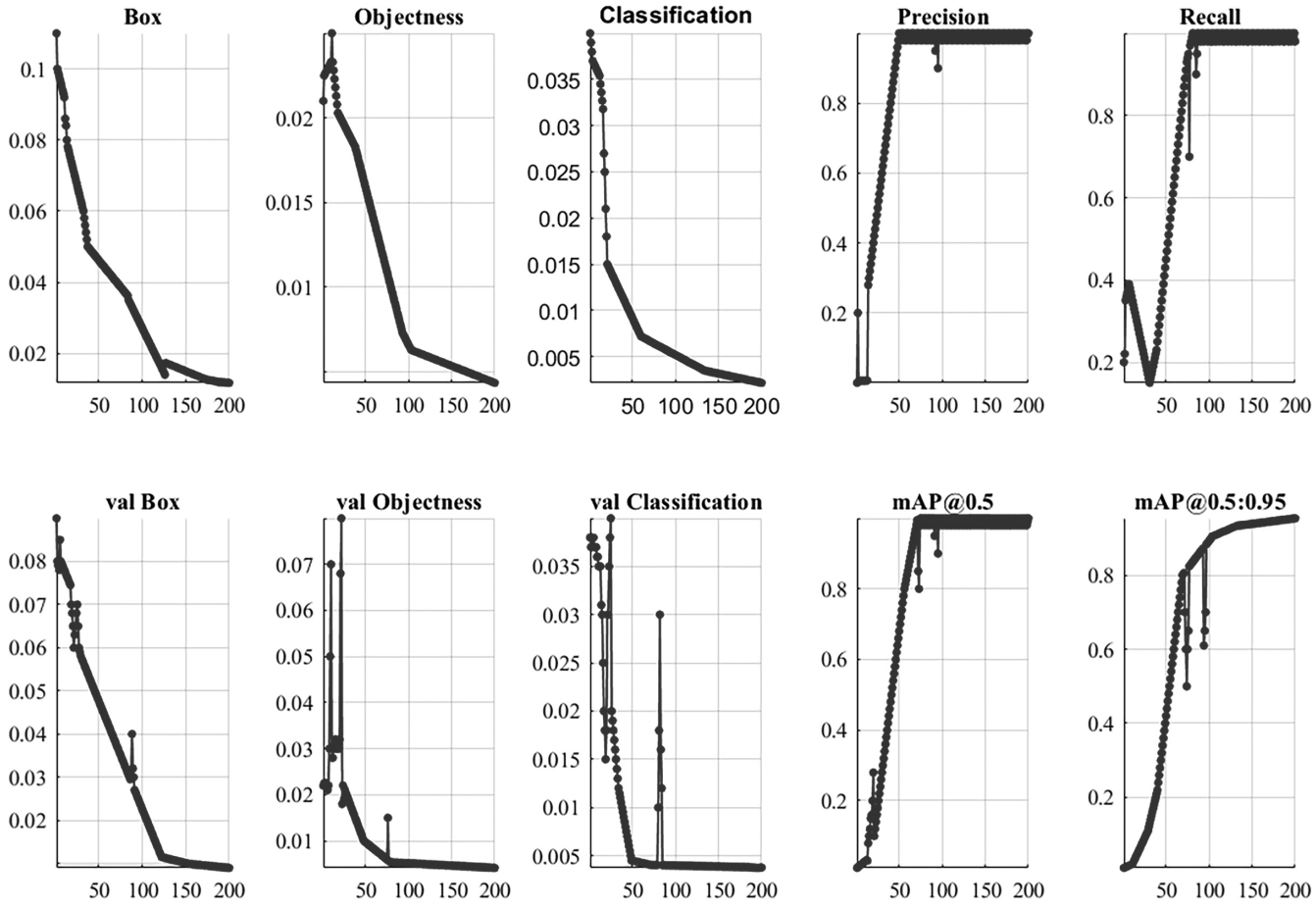


Figure 6. The result of training and validation.

performance of GPU is 114.3 giga floating-point operations per second (GFLOPS). As a result, training 200 epochs providing the highest mAP can be completed in 0.642 h.

5. Experiment

This section presents the experiment in three parts, including result of training and validation, performance of detector, and step of each posture.

5.1 Validation of Training Set

The experiment is to evaluate the performance of the detector constructed by the training process. The testing set consists of images those are and are not in the training set. The results are illustrated as the following information: (a) *Box* is the error of bounding box in training. (b) *Objectness* is essentially a measure of the probability that an object exists in a proposed region of interest (ROI). (c) *Classification* is the problem of determining the category or goal label in training. (d) *Precision* is referred to the number of true positives divided by the total number of positive predictions. (e) *Recall* is referred to the number of correctly classified positives divided by the total number of positives. (f) *Val box* is the error of bounding box in validation. (g) *Val classification* is the problem of determining the category or goal label in validation. (h) *mAP* is calculated by finding average precision (AP) for each class and then average over a number of classes.

The result of training and validation is shown in Fig. 6 (y-axis is percentage and x-axis is the number of epoch). The error of the box was 0.5%. The error of the objectness was 0.05%. The error of the classification was 0.1%. The precision was 100% and recall was 100%. For validation, the error of the val box was 0.05% yet, the error of the objectness was 0% and the error of the classification was 0%. Finally, the mAP is the results of all the classes. The mAP@0.5 was 100% and mAP@0.5:0.95 was 90%.

5.2 Performance of the detector

Precision is the comparison between true positive and false positive. *Confidence* is confidence score of prediction, where 1 is fully confident and 0 is not confident. When this model had given confidence score less than 5%, precision is very low; otherwise, precision is high [see Fig. 7(a)].

Recall is correctness value comparing between true positive and false positive. When recall is 1, the predicted data is 100% correct. When the recall is 0, the predicted data is incorrect. When confidence is less than 90%, recall is 100% [see Fig. 7(b)].

Every value of recall has high precision. F1 score is average value of precision and recall [see Fig. 7(c)]. When

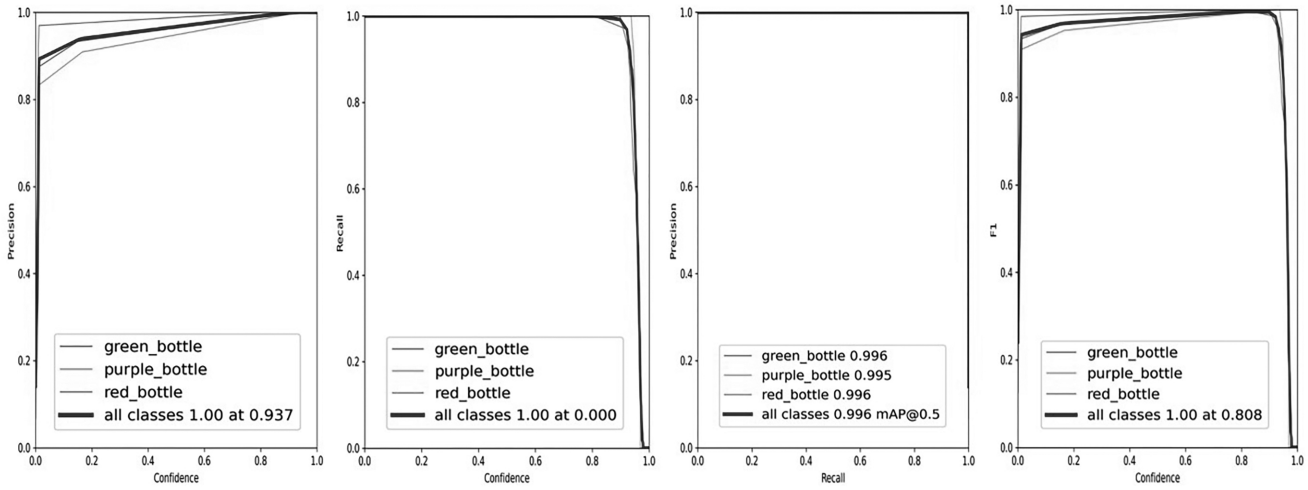
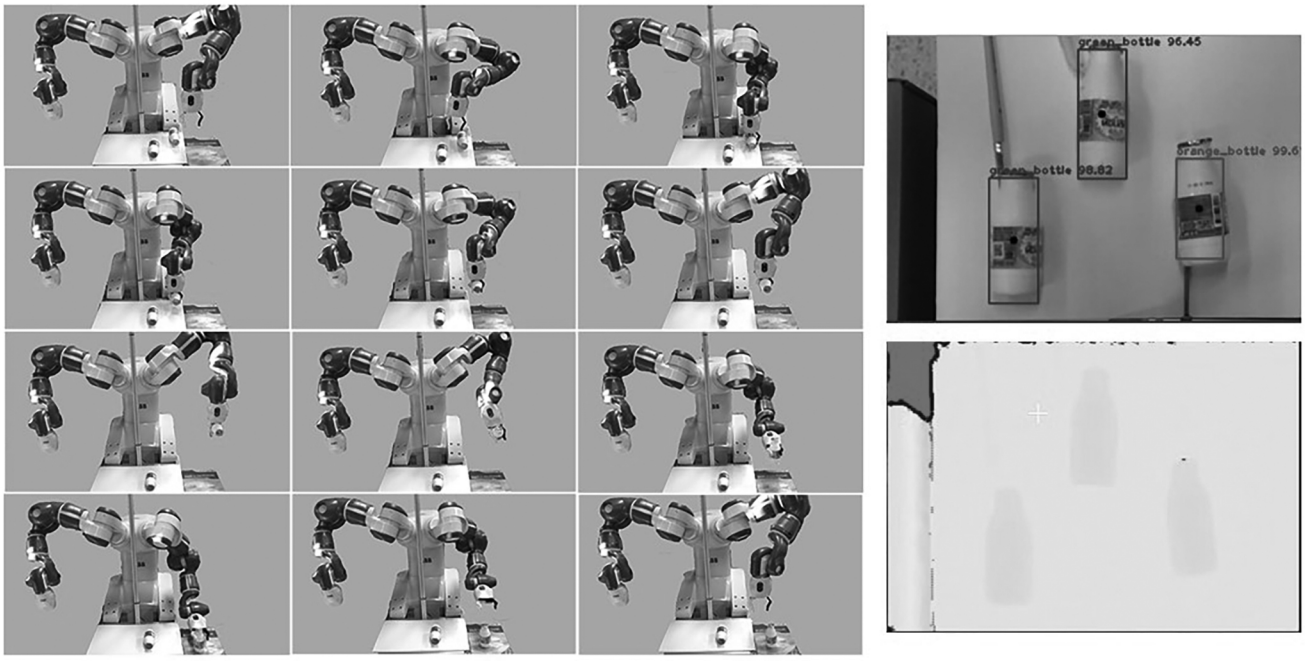


Figure 7. Performance assessment: (a) precision and confidence; (b) recall and confidence; (c) precision and recall; and (d) F1 and confidence.

Table 1
The Steps of Each Posture

Step (Posture) No.	Description	Component Involved (*e.g.*, arm, gripper, and vision, ..)	End Point (x [mm], y [mm], z [mm], Roll [deg], pitch [deg], yaw [deg])	Angle of Joint (j1, j2, j3, j4, j5, j6, j7) [deg]
1	Standby posture before grasp	Arm in standby posture before grasp, vision in real-time detect	–	−30, −52, 50, −257, 49, 31, 59
2	Pre-grasp posture	Arm on the object, gripper open, object is detected	x object, y object, z object + 90, 180, 0, 90	–
3	Grasp posture	Arm on the object, gripper close	x object, y object, z object, 180, 0, 90	–
4	Pre-grasp posture	Arm on the object	x object, y object, z object + 90, 180, 0, 90	–
5	Standby posture before grasp	Arm in standby posture before grasp,	-	−30, −52, 50, −257, 49, 31, 59
6	Standby posture before place	Arm in standby posture before place,	-	0, −80, 51, −270, 42, 83, 65
7,8,9	Pre-place posture	Arm in pre-place posture	320 - offset, 268, 40 + 120, 90, 0, 90	-
10	Place posture	Arm in place posture, gripper open	320 - offset, 268, 40, 90, 0, 90	-
11	Pre-place posture	Arm in pre-place posture	320 - offset, 268, 40 + 120, 90, 0, 90	-
12	Standby posture before grasp	Arm in standby posture before grasp, vision in real-time detect	-	−30, −52, 50, −257, 49, 31, 59

predicted data has confidence between 5% and 95% that give F1 more than 90% [see Fig. 7(d)]. However, precision can be seen as a measure of quality and recall as a measure of quantity. F1 score is the harmonic mean combining recall and precision. It maintains a balance between recall and precision.



(a) (b)

Figure 8. The images are labelled in an alphabetical: (a) the posture of YuMi robot for grasping object. (b) RGB and depth image of bottles detection by YOLOv5.

5.3 Robot Operation

The experiment was done with ten images in two conditions, with and without robot operation, under assumption that YOLOv5 was able to detect the object 100% accuracy. Focusing on the image processing part alone, the average time for classification and localisation is 0.014 s (min = 0.011 max = 0.021 SD = 0.0034). Therefore, the framerate is 71.94 fps which is fast enough for the real-time estimation. However, when realising the robot operation, the average time is 17.32 s (min = 16.88 max = 18.02 SD = 0.429). Little time different was occurred due to different colour and posture of the objects. In Table 1, robot operation took a lot of time due to the physical movement from one position to another. The processing time includes the time for image processing for detection and other computation process.

The operation includes that step involving the robot and vision system. Fig. 8 shows the operation of the robot grasping the bottles and move to the target location. The steps are related to the postures that some of them are required in order to resolve singularity as follows.

5.4 Discussion

(a) *The validation of the training set* shows the performance during the training and validation process of the captured images. The result shows high precision and low error. Meanwhile, the experimental result shows the performance of the detector. (b) *The comparison between precision – confidence – recall* shows the performance is high in every perspective. (c) In regard to *the robot operation*, real-time detection is possible due to the short operation time required. However, the robot operation takes long time in comparison to the detection process alone. (d) About the robot movement, the end-points are set as the target position that the robot will grasp the object. However, the singularity can be presented along the way when moving from certain starting points. Therefore, via points are set as the standby posture in which the movement is done with joint coordinate.

6. Conclusion

A lot of research works currently have implemented industrial robots with vision system using 3D cameras. AI has increased the application in vision system for handling more complex situation, such as detection of objects with variations. In this research, YOLOv5 is implemented with the YuMi dual-arm collaborative robot and Realsense RGB-D camera. The colour images are used for training and detection while the depth image is used for localisation.

From the experiment, the error of the box, objectness, and classification of training and validation were less than 1%, while the precision and recall of training and validation were 100%. This model shows very high performance in object detection (97.4% precision and 98.2% recall). However, the performance can be marginally lower if the ambient conditions, *e.g.*, lighting and shade, affect the objects.

From Fig. 7, it can be implied that the number of training set with a more variation in background and lighting conditions is required to increase performance. However, Fig. 7 shows that the performance can be increased by limiting the confidence in the program between 60% and 90%, which will result in the best outcome.

To validate the performance due to the time consumption, processing time was 0.014 s in average (∼71 fps), which is sufficient for real-time detection. The time for robot operation was 17.32 s in average. It takes a lot time as it needs to avoid the singularity by moving to predefined via points.

In conclusion, using YOLOv5 with YuMi robot could be efficient. Machine learning approach is more flexible than traditional machine vision methods. It achieved high precision, lower error, and high endurance to light variation. This has proved to be sufficient for most industrial applications.

Acknowledgement

We would like to thank The Royal Golden Jubilee Ph.D. Programme (RGJ Ph.D.) contract number PHD/0202/2561, which is the scholarship supported by Thailand Research Fund (TRF). We also would like to thank ABB Automation (Thailand) Co., Ltd. who support the YuMi robot for this research.

References

[1] A. Zakhama, L. Charrabi, and K. Jelassi, Intelligent selective compliance articulated robot arm robot with object recognition in a multi-agent manufacturing system, *International Journal of Advanced Robotic Systems*, *16*(2), 2019, 1729881419841145.

[2] D. Kirschner, R. Velik, S. Yahyanejad, M. Brandstötter, and M. Hofbaur, YuMi, Come and play with me! A collaborative robot for piecing together a tangram puzzle, in A. Cham, G. Rigoll Ronzhin, and R. Meshcheryakov (eds.), Interactive collaborative robotics, (New York: Springer International Publishing, 2016), 243–251.

[3] J. Liang, G. Zhang, W. Wang, Z. Hou, J. Li, X. Wang, and C.-S. Han, Dual quaternion based kinematic control for Yumi dual arm robot, *Proc. 2017 14th International Conf. on Ubiquitous Robots and Ambient Intelligence (URAI)*, 28 June–1 July 2017, 2017, 114–118, doi: 10.1109/URAI.2017.7992899.

[4] S.-H. Wu and X.S. Hong, Integrating computer vision and natural language instruction for collaborative robot human-robot interaction, *Proc. 2020 International Automatic Control Conf. (CACS)*, 4–7 Nov. 2020, 1–5, doi: 10.1109/CACS50047.2020.9289768.

[5] R. Yang, T.P. Nguyen, S.H. Park, and J. Yoon, Automated picking-sorting system for assembling components in an IKEA chair based on the robotic vision system, *International Journal of Computer Integrated Manufacturing*, *35*(6), 2022, 583–597.

[6] P. Opaspilai, S. Vongbunyong, and A. Dheeravongkit, Robotic system for depalletization of pharmaceutical products, *Proc. 2021 7th International Conf. on Engineering, Applied Sciences and Technology (ICEAST)*, Pattaya, 2021, 133–138.

[7] J. Lelachaicharoeanpan and S. Vongbunyong, Classification of surgical devices with artificial neural network approach, *Proc. 2021 7th International Conf. on Engineering, Applied Sciences and Technology (ICEAST)*, Pattaya, 2021, 154–159.

[8] Y. Wang, J. Qiu, J. Wu, and J. Wang, Inverse kinematic solution of a 7-DOF robot with a telescopic forearm based on joint limit and inertia matrix fluctuation, *International Journal of Robotics and Automation*, *38*, 2023, 50–59.

[9] W. Fang, L. Wang, and P. Ren, Tinier-YOLO: A real-time object detection method for constrained environments, *IEEE Access*, *8*, 2020, 1935–1944, doi: 10.1109/ACCESS.2019.2961959.

[10] J. Du, Understanding of object detection based on CNN family and YOLO, *Journal of Physics: Conference Series*, *1004*, 2018, 012029, doi: 10.1088/1742-6596/1004/1/012029.

[11] J. Sang, Z. Wu, P. Guo, H. Hu, H. Xiang, Q. Zhang, and B. Cai, An improved YOLOv2 for vehicle detection, *Sensors*, *18*(12), 2018, 4272.

[12] J. Zhang, M. Huang, X. Jin, and X. Li, A real-time Chinese traffic sign detection algorithm based on modified YOLOv2, *Algorithms*, *10*(4), 2017, 127.

[13] B. Benjdira, T. Khursheed, A. Koubaa, A. Ammar, and K. Ouni, Car detection using unmanned aerial vehicles: comparison between faster R-CNN and YOLOv3, *Proc. 2019 1st International Conf. on Unmanned Vehicle Systems-Oman (UVS)*, Muscat, 5-7 Feb. 2019, 2019, 1–6, doi: 10.1109/UVS.2019.8658300.

[14] R. Girshick, Fast R-CNN, *Proc. of the IEEE International Conf. on Computer Vision*, Santiago, 2015, 1440–1448.

[15] B. Tekin, S.N. Sinha, and P. Fua, Real-time seamless single shot 6D object pose prediction, *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018, 292–301.

[16] L. Zhao and S. Li, Object detection algorithm based on improved YOLOv3, *Electronics*, *9*(3), 2020, 537.

[17] A. Kuznetsova, T. Maleva, and V. Soloviev, Detecting apples in orchards using yolov3 and yolov5 in general and close-up images, in M. Cham, S. Qin Han, and N. Zhang (eds.), Advances in neural networks – ISNN 2020, (New York: Springer International Publishing, 2020), 233–243.

[18] G. Yang, W. Feng, J. Jin, Q. Lei, X. Li, G. Gui, and W. Wang, Face mask recognition system with YOLOV5 based on image recognition, *Proc. 2020 IEEE 6th International Conf. on Computer and Communications (ICCC)*, Chengdu, 2020, 1398–1404.

[19] M. Lai and L. Gao, Automatic classification of apple leaf diseases based on transfer learning, *International Journal of Robotics and Automation*, *37*(1), 2022, 44–51.

[20] D. Dlužnevskij, P. Stefanovic, and S. Ramanauskaite, Investigation of YOLOv5 efficiency in iPhone supported systems, *Baltic Journal of Modern Computing*, 9(3), 2021, 333–344.

Biographies

Dumrongsak Kijdech received the B.Eng. and M.Eng. degrees in mechanical engineering from Rajamangala University of Technology Phra Nakhon, Thailand, in 2011, the M.Eng. degree in mechanical engineering from Kasetsart University, Thailand, in 2015. He is currently pursuing the Ph.D. degree with the Institute of Field Robotics, King Mongkut's University of Technology Thonburi, Thailand. He is currently a Lecturer in Mechanical Engineering with Southeast Asia University, Thailand. His main research directions include artificial intelligence (YOLOv3, YOLOv5, CNN, and Mask R-CNN), automatic control, smart farm, image processing, and robot control.

Supachai Vongbunyong received the B.Eng. and M.Eng. degrees in mechanical engineering from Chulalongkorn University, Thailand, in 2005 and 2009, respectively, and the Ph.D. degree in manufacturing engineering from the University of New South Wales (UNSW), Australia, in 2013. He is currently an Assistant Professor with the Institute of Field Robotics (FIBO) and the Director of King Mongkut's University of Technology Thonburi, Bangkok, Thailand. He started his post-doc research as a Research Associate in 2013 with the School of Mechanical and Manufacturing Engineering, UNSW. His expertise is in robotics and automation. He is a Co-Founder of Innovation and Advanced Manufacturing Research Group and a Founder of Hospital Automation Research Center at FIBO.