

# Hybrid object detection and distance measurement for precision agriculture: integrating YOLOv8 with rice field sidewalk detection algorithm

Anucha Tungkasthan<sup>1</sup>, Nipat Jongsawat<sup>1</sup>, Padma Nyoman Crisnapati<sup>2</sup>, Yamin Thwe<sup>1</sup>

<sup>1</sup>Data and Information Science, Faculty of Science and Technology, Rajamangala University of Technology Thanyaburi, Nakhon Nayok, Thailand

<sup>2</sup>Department of Mechatronics Engineering, Faculty of Technical Education, Rajamangala University of Technology Thanyaburi, Nakhon Nayok, Thailand

## Article Info

### Article history:

Received Jul 8, 2024

Revised Nov 9, 2024

Accepted Nov 14, 2024

### Keywords:

Hybrid approach

Object detection

Precision agriculture

Rice field sidewalk detection

algorithm

YOLOv8

## ABSTRACT

This study proposes an approach to semantic segmentation of sidewalk images in rice fields using hybrid object detection and distance estimation, to enhance agricultural monitoring and analysis. The experimental process involved preparing the development environment, extracting feature vectors and annotations from images, and training the model using YOLOv8. Evaluation reveals consistent and accurate sidewalk detection with a confidence score of 0.9-1.0 across various environmental conditions. Confusion matrix and precision-recall analysis confirmed the robustness and accuracy of the model. These findings validate the effectiveness of the approach in detecting and measuring sidewalks with high precision, potentially improving agricultural monitoring. The novelty of this study lies in the utilization of the rice field sidewalk detection (RIFIS-D) algorithm as an integral part of a hybrid approach with YOLOv8. This hybridization enriches the model with additional capability to detect the distance between the sidewalk and the tractor, addressing specific needs in agricultural applications. This contribution is significant in the advancement of automatic navigation and monitoring technology in agriculture, enabling the implementation of more sophisticated and efficient systems in field operations.

*This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.*



## Corresponding Author:

Nipat Jongsawat

Data and Information Science, Faculty of Science and Technology

Rajamangala University of Technology Thanyaburi

Pathum Thani, Thailand

Email: nipat\_j@rmutt.ac.th

## 1. INTRODUCTION

Obstacle detection is an important element in autonomous robot navigation, with the complex trade-off between safety and mobility at the heart of the problem. The former ensures that the robot does not harm itself or the environment, including humans and animals. The latter is equally crucial, as it enables the robot to fully plan and execute paths, thus determining its ability to successfully complete tasks [1]. Additionally, obstacle detection needs to be updated rapidly, allowing the robot to react promptly to safety-critical information. This need is especially pronounced when transitioning from controlled indoor environments to more challenging outdoor settings, where safe navigation becomes paramount [2]–[4]. In this study, we focus on outdoor environments where robots must navigate various terrains, including roads, paths, grasslands, meadows, and forest trails. This setup presents particular challenges, as the

definition of obstacles varies depending on terrain type. Such considerations are pertinent to a range of robot applications, including forestry and agriculture [5].

Several researchers have conducted studies on object detection across various environments. These include obstacle detection [6], crack detection in tiled sidewalks [7], identification of accessibility problems on sidewalks [8], detection of furrows in corn fields [9], and identification of pathways in rice fields for agricultural vehicles [10]. These studies underscore the importance of real-time detection and adaptability to diverse environmental conditions. Terrain-adaptive obstacle detection integrates 3D-light detection and ranging (LiDAR) data with geometric and semantic terrain features to ensure reliable navigation of autonomous systems across different terrain types [11]. Real-time crack detection in tiled sidewalks utilizes unmanned aerial vehicle (UAV) imagery and YOLO-based methods, demonstrating excellent accuracy and adaptability to environmental factors such as shadows and rain. The PreSight system accelerates object detection by leveraging prior data collection, significantly reducing latency for real-time identification of accessibility issues on sidewalks. Neural network-based algorithms for corn field furrow detection offer high accuracy and versatility, overcoming challenges posed by color and texture similarities. While using LiDAR sensors presents an easy solution, various obstacles, particularly related to cost rather than the sensor itself, may arise [12]–[16]. Thus, this research focuses on developing a rice field sidewalk detection (RIFIS-D) algorithm. The novelty of this research lies in employing a low-cost tool, such as a camera, for detection, as opposed to sensors with environmental mapping capabilities like LiDAR as shown in Figure 1. Given the usage of a camera as a sensor, an accurate detection algorithm becomes imperative. The proposed algorithm incorporates a hybrid technique combining YOLOv8 and RIFIS-D.

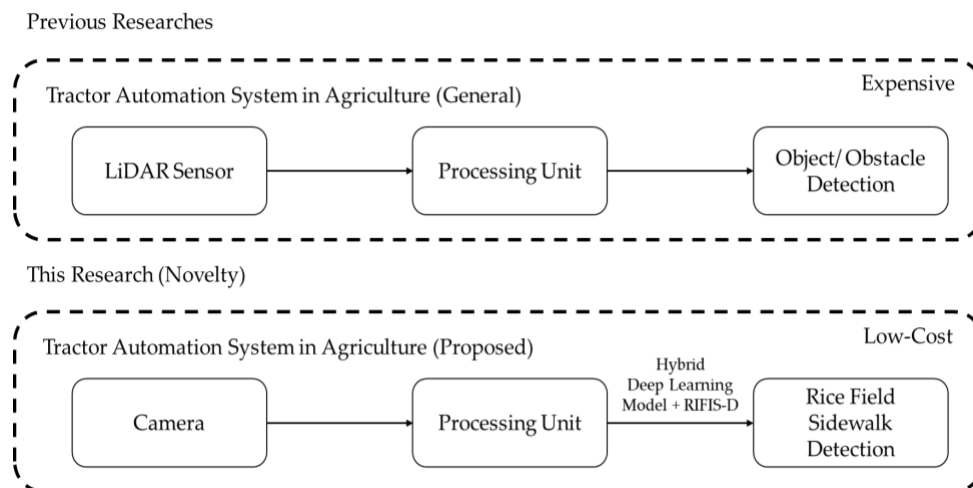


Figure 1. Comparing the novelty of this research with previous studies helps to underscore its unique contributions to the field

The main algorithm used in RIFIS-D Sidewalk detection is based on YOLOv8. YOLOv8, short for YOLOv8, is a deep learning algorithm employed for object detection. It is notable for its speed and accuracy, making it well-suited for real-time applications [17]–[20]. Compared to other object detection algorithms, YOLOv8 offers several advantages, including high accuracy at rapid inference speeds, simultaneous detection of multiple objects, and seamless integration into various systems. These features render it a preferred choice over other algorithms for object detection tasks. However, as this algorithm cannot function in isolation, an additional algorithm in the form of RIFIS-D is necessary to detect the distance between the Sidewalk and the Tractor. The RIFIS-D algorithm involves reading an image using OpenCV, dividing it into chunks with a defined function, and preprocessing it with bilateral filtering and edge detection. Edge array generation identifies edges and divides them for processing. Lines are drawn based on calculated coordinates, connecting points in the edge array and extending from the bottom center of the image. The processed image is then displayed. This process enables efficient RIFIS-D in agricultural images, supporting tasks such as plowing fields with hand tractors. An overview of the concept of this research is provided in Figure 2.

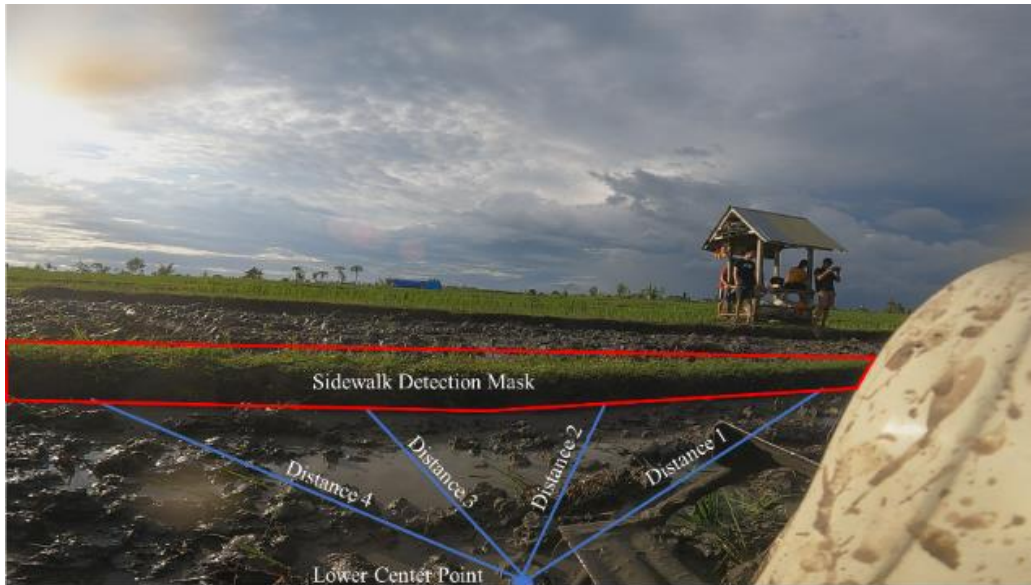


Figure 2. The basic concept of the research

## 2. METHOD

### 2.1. The process of generating the YOLOv8 model

The generation of the YOLOv8 model begins with comprehensive data preparation. This involves collecting relevant images and videos from various sources and meticulously annotating them with accurate bounding boxes and class labels to facilitate supervised learning [21]. To increase dataset variability and robustness, data augmentation techniques such as rotation, flipping, and scaling are applied [22]. The annotated data is then converted into a suitable format, typically involving java script object notation (JSON) or extensible markup language (XML) files, and divided into training, validation, and test subsets to ensure the model's performance is accurately assessed. Preprocessing steps like normalizing and resizing images, along with efficient data loading pipelines, prepare the data for the training phase. Quality control measures are implemented to ensure the dataset is free from errors and inconsistencies [23]. During model training, the process starts with initializing the YOLOv8 model architecture, and defining its layers and parameters. An appropriate loss function, such as mean squared error for regression tasks or cross-entropy loss for classification, is selected to measure the model's performance. The choice of optimizer, such as Adam or stochastic gradient descent (SGD), is crucial for adjusting the model weights during training [24]. The training loop involves iterating over the training data, performing forward and backward passes, and updating the model parameters. Regular checkpointing saves the model's state, allowing training to resume from specific points if necessary. Model testing involves running the trained YOLOv8 model on the test dataset to generate predictions. Post-processing steps, such as non-maximum suppression, refine these predictions. Evaluation metrics include constructing a plotting precision and recall curves to assess the trade-off between precision and recall [25]. Once the YOLOv8 model is trained and tested, it is deployed for RIFIS-D.

### 2.2. The process of RIFIS-D

The RIFIS-D process leverages the trained YOLOv8 model as its core component. Initially, dataset validation images are processed using the YOLOv8 model to ensure data quality and the model's detection accuracy. The trained model is then used to detect objects within the images provided by the rice field sidewalk (RIFIS) system. The RIFIS-D algorithm begins by reading the input image and, if necessary, dividing it into manageable chunks to enhance processing efficiency. Pre-processing steps are performed on the image chunks, such as normalization and noise reduction, followed by segmenting the image into different areas for focused analysis and drawing a center line to assist in spatial orientation. The final stage involves robot movement based on the detection results. The robot is commanded to move forward, turn left, or turn right depending on the presence and location of obstacles or targets detected by the model. The robot is instructed to stop when necessary, such as upon reaching the target or detecting an obstruction. This structured methodology ensures that the YOLOv8 model is effectively integrated into the RIFIS-D system, enabling accurate object detection and responsive robot movement. Figure 3 illustrates the detailed step-by-step process of the research flow conducted in this study.

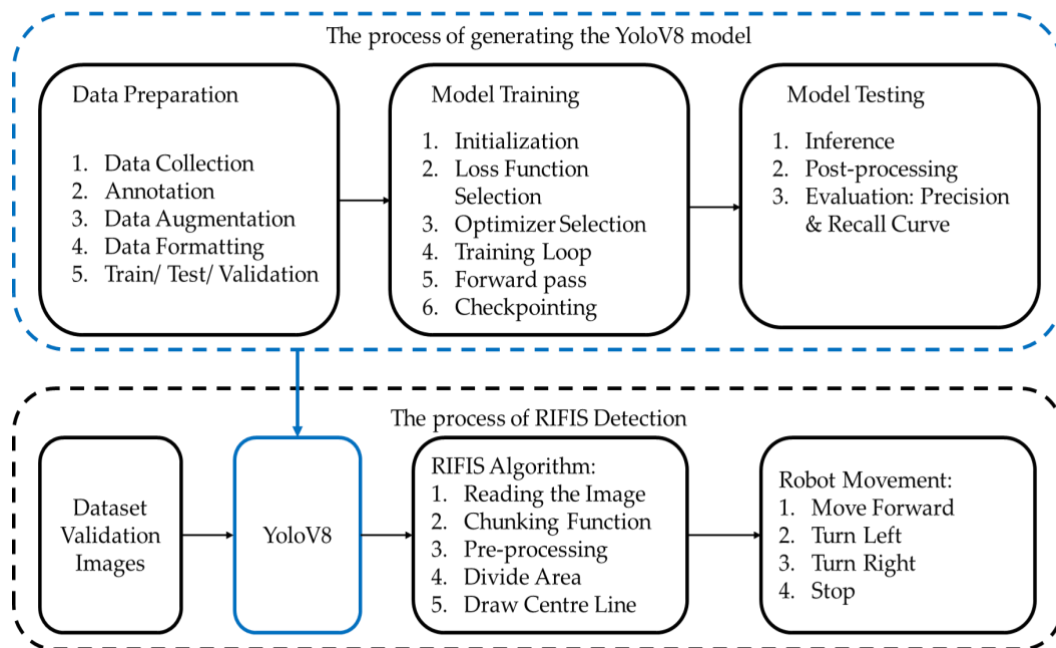


Figure 3. Research flow and proposed hybrid algorithm

### 2.3. Data preparation

The data used in this research is sourced from an open dataset available at [26]. This dataset comprises videos capturing farmers plowing their fields using a tractor, with three cameras positioned on the right, left, and front sides of the tractor. From this dataset, 866 frames were selected for use as training and testing datasets, based on specific observations. Subsequently, an annotation process was conducted using Roboflow [27]. The dataset images were segmented into three parts: the rice field area, sidewalk, and area outside the rice field. This segmentation resulted in six ground truths, serving as references for labeling images/annotations. Of the 866 frames/images selected, augmentation was performed using the color jitter technique, involving random adjustments to brightness, contrast, saturation, and hue values. This process yielded a total of 1732 images. Figure 4 illustrates ground truth region of interest (RoI) annotations for the dataset, showcasing various point configurations utilized to accurately delineate objects for YOLOv8 model training. Each subfigure represents a different number of annotation points, tailored to the complexity and shape of the object: Figure 4(a) six points for detailed contours, Figure 4(b) seven points for irregular shapes, Figure 4(c) five points for a balance between detail and simplicity, Figure 4(d) four points for rectangular objects, Figure 4(e) three points for simpler or triangular objects, and Figure 4(f) eight points for highly complex shapes. The legend as shown in Figure 4(g) provides a key for interpreting the symbols and color codes used in this annotation.

Next, the annotation format is determined in JSON form, which includes a comprehensive structure for organizing image and annotation data. Each entry in the image list contains details about an image, such as the file name, height, width, and a list of annotations. Each annotation specifies a class ID, class name, and bounding box coordinates (in [x, y, width, height] format), representing the top-left corner and dimensions of the bounding box. Additionally, the JSON includes a 'sidewalk' category, defining different classes of objects with their respective IDs and names. This structured format ensures that all the information required for YOLOv8 model training is clearly organized, facilitating efficient data handling and model training. The dataset of 1,732 images were divided into three subsets to ensure training, validation, and testing of the YOLOv8 model. Specifically, 70% of the dataset (1,212 images) is allocated for training to provide the model with enough data to learn patterns and features. Another 15%, or 260 images, are set aside for validation, used during training to tune hyperparameters and monitor performance, thus helping to prevent overfitting through techniques like early stopping. The remaining 15%, also 260 images, are intended for testing, enabling an unbiased evaluation of the model's performance on unseen data. This division ensures that the model generalizes well and provides a reliable measure of accuracy and robustness.

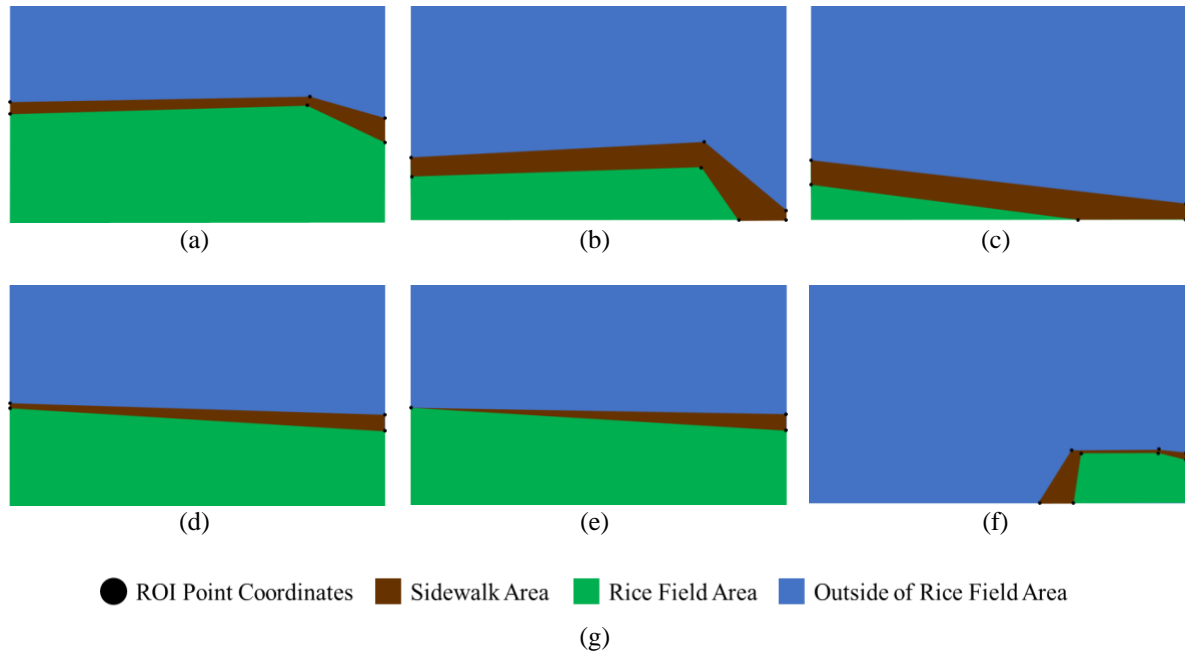


Figure 4. Ground truth for dataset annotations (a) six points, (b) seven points, (c) five points, (d) four points, (e) three points, (f) eight points, (g) legend

### 3. RESULTS AND DISCUSSION

This research proposes an approach to perform semantic segmentation on images of sidewalks in rice fields using the YOLOv8 algorithm. First, the steps for preparing the development environment are explained, including the installation of libraries such as 'supervisely' and 'ultralytics', as well as the utilization of Google Colab and Google Drive to store the dataset. Next, the process of extracting information from the dataset in the form of images and annotations is carried out by reading JSON data and processing each image along with its annotations. Subsequently, segmentation information is written into a text file based on the annotations associated with each image. Following data preparation, the YOLOv8 model is trained for semantic segmentation, utilizing parameters such as a previously prepared model saved in 'yolov8s-seg.pt', data configuration stored in a .yaml file, ten epochs, and an image size of 640×640 pixels.

At the evaluation stage, a review of the training results was conducted, including an analysis of the resulting segmentation images, confusion matrix, and precision-recall metrics. Figure 5 presents the outcomes of sidewalk detection in rice fields using the YOLOv8 algorithm, where each box represents a frame from the video or a set of processed images. Detected sidewalk areas are highlighted in pink and labeled as 'sidewalk' along with a detection confidence score. Sidewalk detection received a confidence score ranging from approximately 0.9 to 1.0, indicating a high level of confidence in the algorithm's performance. Despite variations in angles and lighting conditions across the images, the sidewalk detection remained consistent, demonstrating the robustness of the YOLOv8 algorithm. Detection consistency is excellent, with the majority of frames achieving a confidence score of 1.0. This underscores the YOLOv8 algorithm's capability to accurately and consistently detect sidewalks across various image conditions, offering potential applications in agricultural analysis and land monitoring.

Figure 6 consists of graphs illustrating the results from training the YOLOv8 model, offering a comprehensive depiction of the development of loss and performance metrics throughout the training process. The 'train/box\_loss' graph exhibits a consistent decrease in bounding box prediction loss during each training epoch, with a reduction from 6.5 to approximately 2.5. Similarly, 'train/seg\_loss' and 'train/cls\_loss' display a steady decline in segmentation and classification prediction loss, with 'train/seg\_loss' decreasing from around 4.5 to 1.5, and 'train/cls\_loss' dropping from 4.0 to 1.2. The graphs 'metrics/precision(B)' and 'metrics/recall(B)' demonstrate a notable increase in precision and recall for class 'B', with precision rising from about 0.5 to 0.9, and recall increasing from 0.4 to 0.8. Moreover, 'metrics/mAP50(B)' exhibits a significant enhancement in mean average precision (mAP) for the class, escalating from approximately 0.3 to 0.7. Similar improvements are observed in class 'M', with precision, recall, and mAP50 showing consistent increases as the model is trained. The validation graphs, 'val/box\_loss', 'val/seg\_loss', 'val/cls\_loss', and 'val/df\_loss', also demonstrate a consistent reduction in loss, with each reaching lower values compared to

the training graphs. Additionally, 'metrics/mAP50(M)' and 'metrics/mAP50-95(M)' display substantial increases in mAP for class 'M' throughout the training process, indicating enhanced precision and recall at various intersection over union (IoU) thresholds. Overall, these graphs provide a clear depiction of the model's progression in reducing loss and improving performance metrics during training.

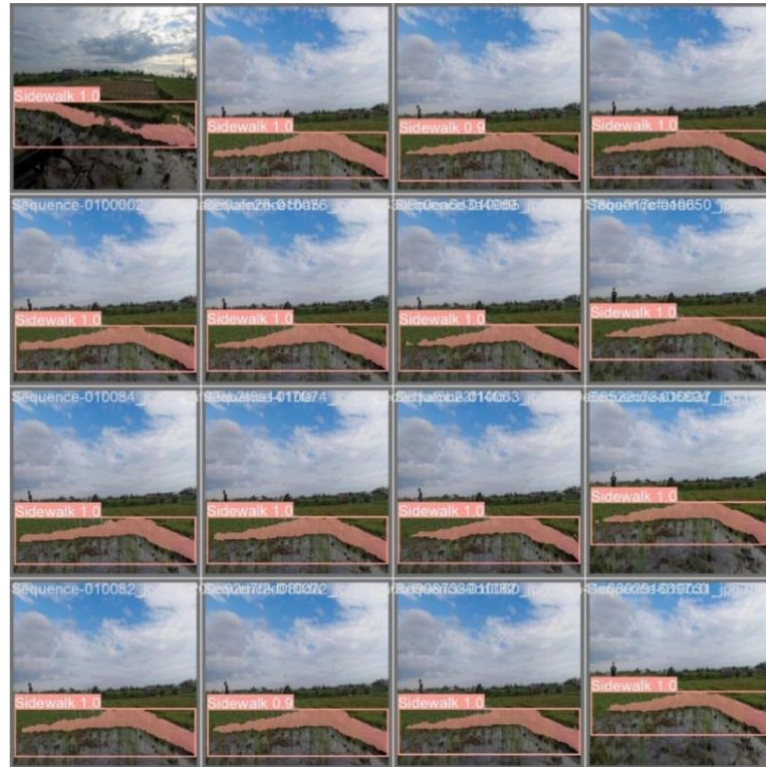


Figure 5. Sidewalk detection results using YOLOv8

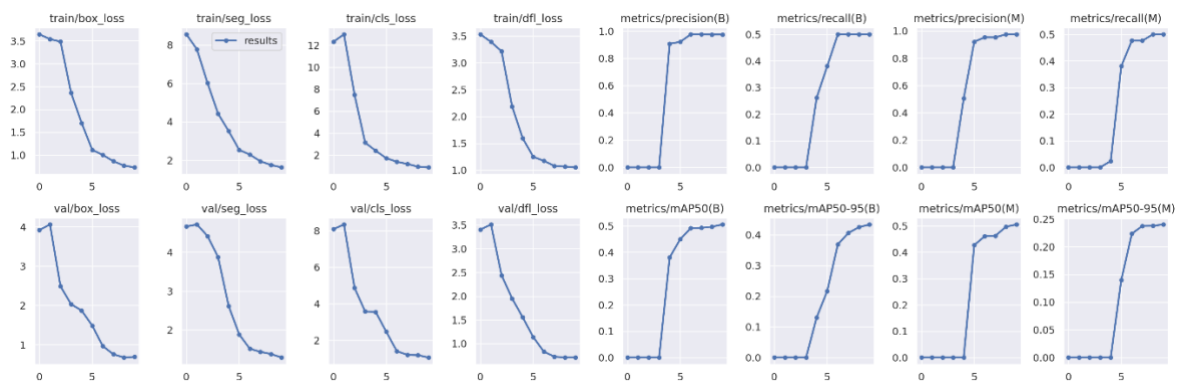


Figure 6. Collection of graphs of model training results using YOLOv8

Utilizing the YOLOv8 model generated and saved in 'best.pt' format, a mask extraction process is initiated from the detected sidewalks. Initially, the pre-trained YOLOv8 model is loaded, followed by the execution of inference on the specified image files, yielding a list of segmentation results. Subsequently, the process iterates through each segmentation result, extracting the mask tensor and converting it into a NumPy array. For each mask, a corresponding Python imaging library (PIL) image object is instantiated, converting the pixel values to 'uint8' format and scaling them within the range [0, 255]. Eventually, the resulting mask image is saved as a '.jpg' file. This methodology ensures the accurate preservation of

segmentation masks extracted from the YOLOv8n-seg model as '.jpg' images, thereby facilitating the subsequent distance measurement process using the RIFIS-D algorithm approach.

RIFIS-D is an algorithm proposed in this research based on input from previous sidewalk detections. To provide further clarification of this algorithm, several mathematical formulas are presented. The first formula describes the image processing process to detect edges and calculate a certain distance. Firstly, the original image, denoted as  $I$ , undergoes application of a bilateral filter to smooth the image, resulting in  $I_{blur}$  as shown in (1). Subsequently, edge detection using the Canny method is performed on the blurred image, generating an edge map  $I_{edge}$  as presented in (2). Following this, in the vertical edge detection step, a vertical scan is conducted on the edge map with a step size  $S$  to identify the lowest edge coordinate in each column, which is stored in  $E_{array}$  according to (3). This array is then partitioned into three parts, denoted as  $slc$  in (4). The average coordinates  $avg_x$  and  $avg_y$  are calculated for each part, as outlined in (5), and the line distance is measured from the bottom center point of the image to the average point using the Euclidean formula, yielding  $l_{line}$  in (6). Finally, lines are drawn between the detected pairs of edge points, as well as lines from the bottom of the image to each edge point, and the results are displayed in the final image. Details of each symbol can be found in Table 1.

$$I_{blur} = \text{bilateralFilter}(I, d = 9, S_c = 40, S_s = 40) \quad (1)$$

$$I_{edge} = \text{Canny}(I_{blur}, t_1 = 50, t_2 = 100) \quad (2)$$

$$E_{array} = \{(j, i) | i \in [H - S], I_{edge}(i, j)\} \quad (3)$$

$$slc = \{E_{array}[k:k+n] | k \in [0, \text{length}(E_{array})], n = \frac{\text{length}(E_{array})}{3}\} \quad (4)$$

$$avg_x = \frac{\sum x_{vals}}{\text{length}(x_{vals})}, avg_y = \frac{\sum y_{vals}}{\text{length}(y_{vals})} \quad (5)$$

$$l_{line} = \sqrt{\left(avg_x - \frac{W}{2}\right)^2 + (avg_y - H)^2} \quad (6)$$

Table 1. Nomenclature

| Symbols     | Description   | Symbols    | Description   |
|-------------|---|------------|---|
| $I$         | : Original image with dimensions H×W (height and width)     | $slc$      | : Slice of $E_{array}$ with size length $\frac{\text{length}(E_{array})}{3}$                              |
| $I_{blur}$  | : Image after application of bilateral filters              | $x_{vals}$ | : List of x coordinates of each slice   |
| $I_{edge}$  | : Edge map resulting from edge detection (Canny method)     | $y_{vals}$ | : List of y coordinates of each slice   |
| $S$         | : Step size (Step Size), here S=5                           | $avg_x$    | : Average x coordinate in one slice   |
| $H$         | : Image height reduced by 1 ( $\text{height}(I)-1$ )        | $avg_y$    | : Average y coordinate in one slice   |
| $W$         | : Image width reduced by 1 ( $\text{width}(I)-1$ )          | $l_{line}$ | : Length of the line measured from the bottom center point of the image to the average point in the slice |
| $E_{array}$ | : Array storing the coordinates of the detected edge points | $S_c$      | : Sigma space   |
| $S_c$       | : Sigma color   |            |   |

Figure 7 demonstrates the creation and testing process of the proposed algorithm through three testing stages. Figure 7(a) depicts the fundamental concept using an ideal image, which undergoes a sequence of operations including smoothing (blurring), edge detection, and distance calculation. Figure 7(b) assesses the algorithm's performance using indoor laboratory images, following the same stages of processing: original image, smoothing, edge detection, and distance calculation. Figure 7(c) evaluates the algorithm using real outdoor images, incorporating object detection using YOLOv8, followed by edge detection and distance calculation. These three parts collectively showcase the algorithm's capability to accurately process images from diverse conditions, enabling precise edge detection and distance calculation.

Figure 8 illustrates the final results of the RIFIS-D algorithm, displaying the distance measurement between the center point of the bottom of the image and the detected sidewalk. The edge of the sidewalk, marked in red, is detected in the image. The algorithm identifies three main points on the sidewalk, with each annotated with the distance from the bottom center point of the image. The blue line represents the distance measurement from the center point of the bottom of the image to the three edge points of the sidewalk. The numbers in the image, such as 353.36, 253.02, and 372.03, denote the Euclidean distance (in pixels) from the bottom center point of the image to each detected edge point. These results underscore the RIFIS-D algorithm's ability to visually and accurately identify objects in images and measure their distances.

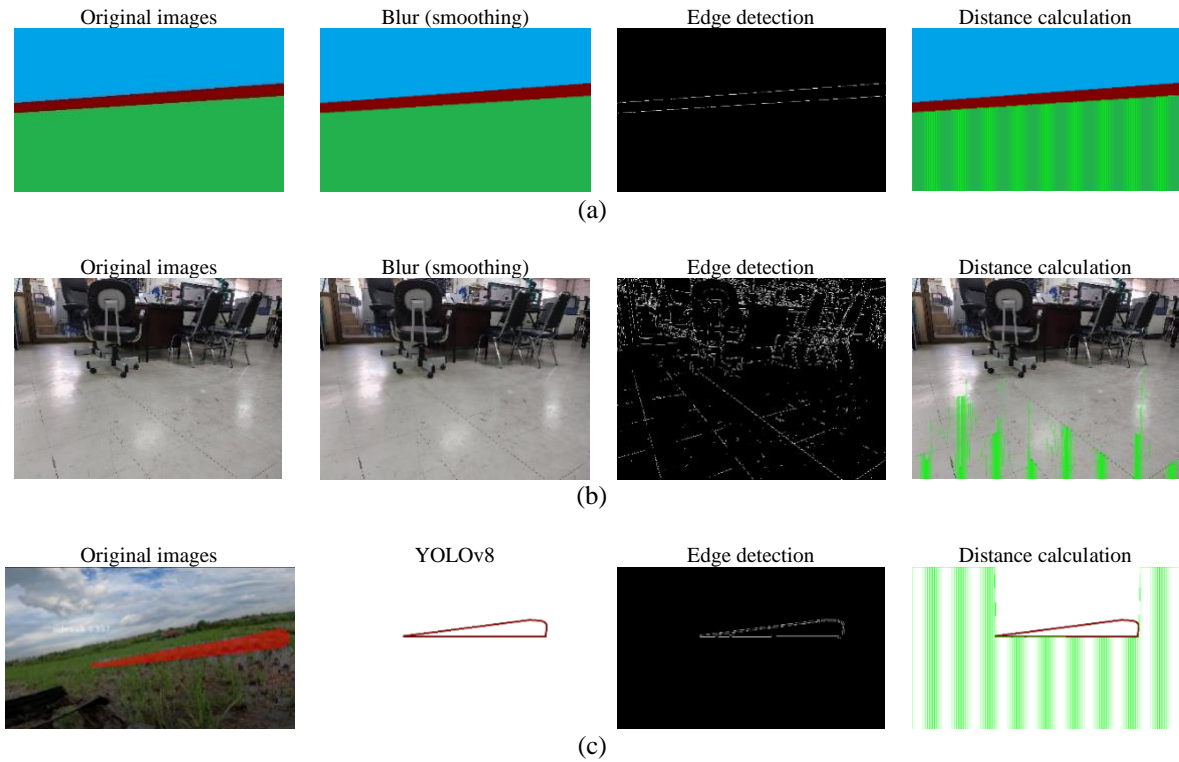


Figure 7. Creation and testing of proposed algorithm; (a) basic concept of using ideal images, (b) algorithm testing using indoor laboratory images, and (c) algorithm testing using original images and detection results by YOLOv8

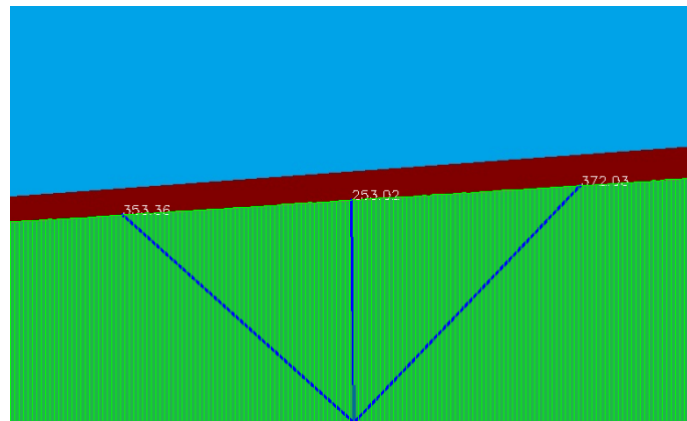


Figure 8. The final result of the RIFIS-D algorithm displays the results of measuring the distance between the center point of the bottom of the image and the detected sidewalk

Figure 9 shows the sidewalk detection process using YOLOv8 as well as the mask extraction and distance measurement steps. In Figure 9(a), the results of sidewalk detection are shown in two different images with YOLOv8, where the areas detected as sidewalks are colored red with confidence levels of 0.92 and 0.95 respectively. The next step is mask extraction, as seen in Figure 9(b), where the detected sidewalk area is represented in binary form (black and white), with white indicating the sidewalk area. The next process is the distance measurement shown in Figure 9(c), where the vertical green lines and blue lines indicate the distance measurement points on the mask, with distance values listed at several points. This process illustrates how YOLOv8 can be used to detect objects accurately and how the detection results can be further analyzed for specific purposes such as measuring the distance between points in the detected area.



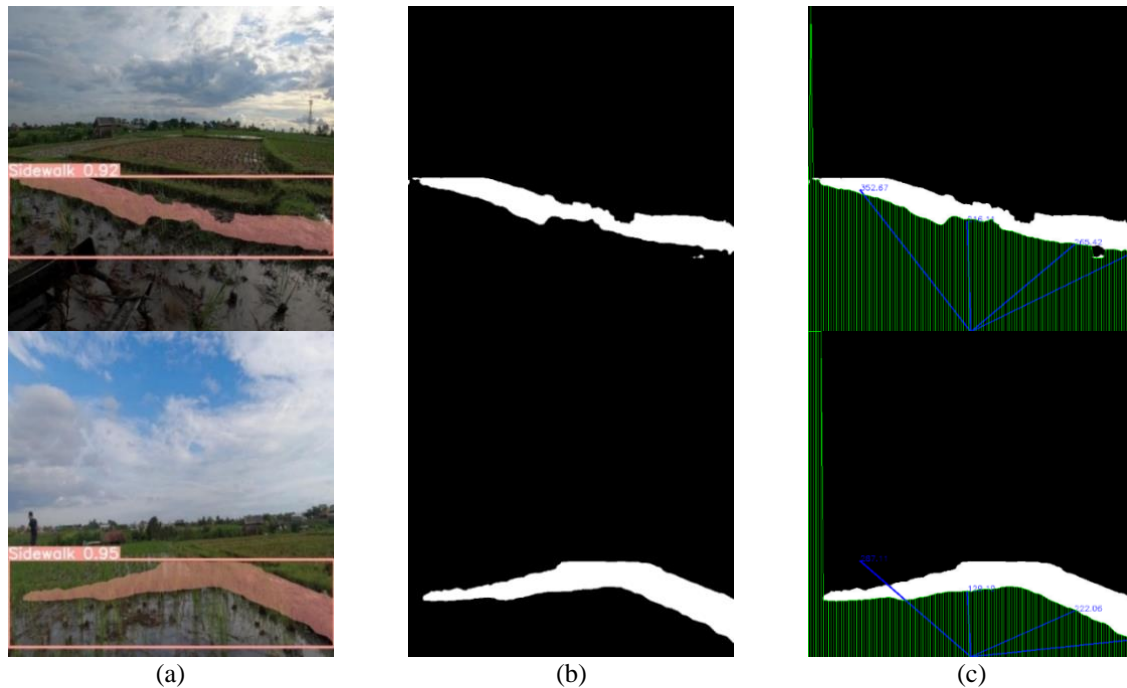


Figure 9. Experimental results; (a) sidewalk detection results using Yolov8, (b) mask extraction, and (c) distance measurement algorithm

#### 4. CONCLUSION

This research describes in detail a new approach for semantic segmentation of sidewalks in rice fields using the YOLOv8 algorithm enriched with the RIFIS-D algorithm. Through a series of experimental steps including development environment preparation, data extraction, model training, evaluation, and analysis, this research succeeded in producing significant findings. The evaluation results show that the proposed model is able to detect and measure sidewalks with high precision, even in a variety of different environmental conditions. It was found that this hybrid approach has consistent robustness and accuracy, and has great potential to improve the efficiency and effectiveness of agricultural monitoring. Thus, this research effectively fills the knowledge gap in the domain of automated navigation and monitoring in agriculture, making a significant contribution to scientific and technological progress in this field. For the future, experimental suggestions include further trials to evaluate the reliability and adaptability of this model in a wider range of environmental scenarios and agricultural applications, while continuing to develop the integration of this technology into more integrated and automated agricultural systems. Thus, this conclusion reflects the detailed findings of the report as well as providing insight into future research directions regarding direct implementation of this algorithm on tractors.

#### ACKNOWLEDGEMENTS

The authors would like to thank RMUTT for the support and facilities they provided for this research. The availability of datasets and computing infrastructure has greatly helped in the development and implementation of the proposed algorithm. The authors also appreciate the collaboration and discussions of the researchers at RMUTT, who provided valuable insights and technical support throughout the research process. The contribution from RMUTT has been an important pillar in the success of this research, and the authors hope that this good collaboration can continue in the future for further research.





#### REFERENCES

- [1] J. P. A. Yaacoub, H. N. Noura, and B. Piranda, "The internet of modular robotic things: Issues, limitations, challenges, & solutions," *Internet of Things*, vol. 23, 2023, doi: 10.1016/j.iot.2023.100886.
- [2] M. Sadaf *et al.*, "Connected and automated vehicles: infrastructure, applications, security, critical challenges, and future aspects," *Technologies*, vol. 11, no. 5, 2023, doi: 10.3390/technologies11050117.
- [3] D. Maneetham, P. N. Crisnapati, and Y. Thwe, "Autonomous open-source electric wheelchair platform with internet-of-things and proportional-integral-derivative control," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 6, pp. 6764–6777, 2023, doi: 10.11591/ijece.v13i6.pp6764-6777.




- [4] S. Ketsayom, D. Maneetham, and P. N. Crisnapati, "AGV maneuverability simulation and design based on pure pursuit algorithm with obstacle avoidance," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 34, no. 2, pp. 835–847, 2024, doi: 10.11591/ijeecs.v34.i2.pp835-847.
- [5] L. Wijayathunga, A. Rassau, and D. Chai, "Challenges and solutions for autonomous ground robot scene understanding and navigation in unstructured outdoor environments: a review," *Applied Sciences*, vol. 13, no. 17, 2023, doi: 10.3390/app13179877.
- [6] J. W. Hu *et al.*, "A survey on multi-sensor fusion based obstacle detection for intelligent ground vehicles in off-road environments," *Frontiers of Information Technology and Electronic Engineering*, vol. 21, no. 5, pp. 675–692, 2020, doi: 10.1631/FITEE.1900518.
- [7] Q. Qiu and D. Lau, "Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images," *Automation in Construction*, vol. 147, 2023, doi: 10.1016/j.autcon.2023.104745.
- [8] C. Bennett, E. Ackerman, B. Fan, J. Bigham, P. Carrington, and S. Fox, "Accessibility and the Crowded Sidewalk: Micromobility's Impact on Public Space," *DIS 2021-Proceedings of the 2021 ACM Designing Interactive Systems Conference: Nowhere and Everywhere*, pp. 365–380, 2021, doi: 10.1145/3461778.3462065.
- [9] N. A. Simon and C. H. Min, "Neural network based corn field furrow detection for autonomous navigation in agriculture vehicles," in *IEMTRONICS 2020- International IOT, Electronics and Mechatronics Conference, Proceedings*, 2020, doi: 10.1109/IEMTRONICS51293.2020.9216347.
- [10] P. N. Crisnapati and D. Maneetham, "Two-dimensional path planning platform for autonomous walk behind hand tractor," *Agriculture*, vol. 12, no. 12, 2022, doi: 10.3390/agriculture12122051.
- [11] S. Jiang, W. Jiang, and L. Wang, "Unmanned aerial vehicle-based photogrammetric 3D mapping: a survey of techniques, applications, and challenges," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 2, pp. 135–171, 2022, doi: 10.1109/MGRS.2021.3122248.
- [12] T. Raj, F. H. Hashim, A. B. Huddin, M. F. Ibrahim, and A. Hussain, "A survey on LiDAR scanning mechanisms," *Electronics*, vol. 9, no. 5, 2020, doi: 10.3390/electronics9050741.
- [13] P. Borges *et al.*, "A survey on terrain traversability analysis for autonomous ground vehicles: methods, sensors, and challenges," *Field Robotics*, vol. 2, no. 1, pp. 1567–1627, 2022, doi: 10.55417/fr.2022049.
- [14] Z. Li, C. Jiang, X. Gu, Y. Xu, Feng zhou, and J. Cui, "Collaborative positioning for swarms: A brief survey of vision, LiDAR and wireless sensors based methods," *Defence Technology*, vol. 33, pp. 475–493, 2024, doi: 10.1016/j.dt.2023.05.013.
- [15] R. Hasan and R. Hasan, "Pedestrian safety using the Internet of Things and sensors: Issues, challenges, and open problems," *Future Generation Computer Systems*, vol. 134, pp. 187–203, 2022, doi: 10.1016/j.future.2022.03.036.
- [16] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39830–39846, 2020, doi: 10.1109/ACCESS.2020.2975643.
- [17] B. Xiao, M. Nguyen, and W. Q. Yan, "Fruit ripeness identification using YOLOv8 model," *Multimedia Tools and Applications*, vol. 83, no. 9, pp. 28039–28056, 2024, doi: 10.1007/s11042-023-16570-9.
- [18] G. Wang, Y. Chen, P. An, H. Hong, J. Hu, and T. Huang, "UAV-YOLOv8: A small-object-detection model based on improved YOLOv8 for UAV aerial photography scenarios," *Sensors*, vol. 23, no. 16, 2023, doi: 10.3390/s23167190.
- [19] J. Tervén, D. M. C. Esparza, and J. A. R. González, "A comprehensive review of YOLO architectures in computer vision: from YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023, doi: 10.3390/make5040083.
- [20] M. Safaldin, N. Zaghdien, and M. Mejdoub, "An improved YOLOv8 to detect moving objects," *IEEE Access*, vol. 12, pp. 59782–59806, 2024, doi: 10.1109/ACCESS.2024.3393835.
- [21] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2021, doi: 10.1016/j.procs.2022.01.135.
- [22] K. Maharana, S. Mondal, and B. Nemade, "A review: data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, 2022, doi: 10.1016/j.gltp.2022.04.020.
- [23] W. Li, M. I. Solihin, and H. A. Nugroho, "RCA: YOLOv8-based surface defects detection on the inner wall of cylindrical high-precision parts," *Arabian Journal for Science and Engineering*, vol. 49, no. 9, pp. 12771–12789, 2024, doi: 10.1007/s13369-023-08483-4.
- [24] M. Reyad, A. M. Sarhan, and M. Arafa, "A modified Adam algorithm for deep neural network optimization," *Neural Computing and Applications*, vol. 35, no. 23, pp. 17095–17112, 2023, doi: 10.1007/s00521-023-08568-z.
- [25] E. Casas, L. Ramos, C. Romero, and F. R. Echeverría, "A comparative study of YOLOv5 and YOLOv8 for corrosion segmentation tasks in metal surfaces," *Array*, vol. 22, 2024, doi: 10.1016/j.array.2024.100351.
- [26] P. N. Crisnapati and D. Maneetham, "RIFIS: A novel rice field sidewalk detection dataset for walk-behind hand tractor," *Data*, vol. 7, no. 10, 2022, doi: 10.3390/data7100135.
- [27] Q. Lin, G. Ye, J. Wang, and H. Liu, "RoboFlow: a data-centric workflow management system for developing AI-enhanced robots," in *Proceedings of Machine Learning Research*, 2021, pp. 1789–1794.

## BIOGRAPHIES OF AUTHORS






**Anucha Tungkasthan**     received a Ph.D. degree in information technology in business (in cooperation with the University of Pittsburgh, USA) from Siam University, Bangkok, Thailand, in 2012 and an M.S. degree in computer education from King Mongkut's University of Technology North Bangkok, Thailand, in 2004. He is currently an Assistant Professor with the Department of Computer Technology, Faculty of Science and Technology, Rajamangala University of Technology Thanyaburi. His research interest includes object detection and tracking, real-time image processing, content-based image retrieval, machine learning, and deep learning. His awards and honors include The Certification of Merit for The World Congress on Engineering, 2010, and The Outstanding Thesis Award of the Association of Private Higher Education Institutions of Thailand. He can be contacted at email: [anucha\\_t@rmutt.ac.th](mailto:anucha_t@rmutt.ac.th).






**Nipat Jongsawat**    was born in Bangkok, Thailand, in 1977. He received a B.S. degree in electrical engineering and M.S. in computer information systems from Assumption University in 1999 and 2002, respectively. He received a Ph.D. degree in information technology in business from Siam University, Thailand, in 2011. He has been an Assistant Professor with the Department of Mathematics and Computer Science, Faculty of Science and Technology, Rajamangala University of Technology Thanyaburi. He has been serving as the faculty's dean since 2018. He is the author of more than 40 articles and 3 book chapters. His research interests include artificial intelligence, collaborative computing, human-computer interaction, decision support systems, group decision support system, group decision-making, computer-supported collaborative learning, computer-supported cooperative work, and business data processing. He is an associate editor of the Journal Progress in Applied Science and Technology. He can be contacted at email: nipat\_j@rmutt.ac.th.



**Padma Nyoman Crisnapati**    after obtaining a Bachelor's degree in 2009 from the Department of Informatics Engineering at Sepuluh Nopember Institute of Technology. He pursued a master's degree in learning technology in 2011 and a master's degree in computer science in 2018 from Ganesha Education University. He is a lecturer at STIKOM Bali Institute of Technology and Business, teaching sensors transducer, assembly language, and animation. He previously served as the Head of the Computer Systems Study Program from 2016 to 2020. He is pursuing a Ph.D. in the Department of Mechatronics Engineering at Rajamangala University of Technology Thanyaburi (RMUTT). His research interests encompass 2D and 3D animation, the internet of things, robotics, automation, augmented and virtual reality, and robotics. He can be contacted at email: crisnapati@rmutt.ac.th.



**Yamin Thwe**    is a dedicated academic and researcher currently pursuing a doctor of engineering in mechatronics engineering at Rajamangala University of Technology Thanyaburi (RMUTT), Thailand, where she maintains an exemplary academic record. Her research interests span across brain signal processing, computer vision, machine learning, big data management, and the internet of things. With a strong background in information technology, including a bachelor of engineering and a master in data and information science, she has made significant contributions to various research projects and publications, particularly in the areas of deep learning and data analysis. She can be contacted at email: yamin\_t@mail.rmutt.ac.th.