An Intelligent Autonomous Document Mobile Delivery Robot Using Deep Learning

https://doi.org/10.3991/ijim.v16i21.32071

Thittaporn Ganokratanaa[1], Mahasak Ketcham[2](✉)
[1] Faculty of Science, King Mongkut's University of Technology Thonburi, Bangkok, Thailand
[2] Faculty of Information Technology and Digital Innovation, King Mongkut's University of Technology North Bangkok, Bangkok, Thailand
mahasak.k@itd.kmutnb.ac.th

Abstract—This paper presents an intelligent autonomous document mobile delivery robot using a deep learning approach. The robot is built as a prototype for document delivery service for use in offices. It can adaptively move across different surfaces, such as terrazzo, canvas, and wooden. In this work, we introduce a convolutional neural network (CNN) to recognize the traffic lanes and the stop signs with the assumption that all surfaces have identical traffic lanes. We train the model using a custom indoor traffic lane and stop sign dataset with the label of motion directions. CNN extracts a direction-of-motion feature to estimate the robot's direction and to stop the robot based on an input image monocular camera view. These predictions are used to adjust the robot's direction and speed. The experimental results show that this robot can move across different surfaces along with the same structured traffic lanes, achieving the model accuracy of 96.31%. The proposed robot helps to facilitate document delivery for office workers, allowing them to work on other tasks more efficiently.

Keywords—autonomous, mobile, delivery robot, convolutional neural network

1 Introduction

Advanced technology plays an important role in the daily lives of humans for all generations. A robot is an advanced technology that is widely used in factories in various systems. It has gained popularity for use with logistics systems in production processes. Transportation is a key function of the logistics system where it transfers goods from the place of origin to another place regarding customer demands at a time. Besides, robot technologies have been implemented in the transportation system by replacing human labor to reduce costs and time in warehouse transfers.

To facilitate the transportation system, automated guided vehicles (AGVs) have been used in modern factory systems for over a decade [1]. The reason why AGVs are important is that they can move around in restricted environments [2-3]. The processing of AGVs can be based on computer vision techniques along with automatic guidance and landmark recognition systems. However, the environment changes, including the brightness of lights, floor patterns, and landmark position, can affect the vision systems

[4,5]. Recent works try to improve the efficiency of the robots by reducing unnecessary movement between shelves and using linear motors to reduce the control complexity. Some works install robots in a factory to move cartons and pallets and to unload and load goods [6-9]. To ensure safe operation, a cloud-based communication system [10,11] was developed to enable communication among autonomous vehicles carrying out transportation functions within a factory. A safe laser scanner or wireless communication system is used to reduce the possibility of collisions [12,13]. Each AGV detects obstacles in the surrounding areas and then either stops or changes its direction of movement to avoid collisions [14,15]. The computer vision techniques used in autonomous robots are important for object detection [16,17] in conjunction with a human operator. Computer vision techniques have been used in various models of autonomous robots, such as road data collection [18-24] and target tracking [25-28]. However, although using autonomous robots to help manage an inventory system is beneficial, path planning is also required for this task. Path planning algorithms have been well-developed and improved to be more efficient; some examples include the field D* algorithm and the improved chaotic motion path planner [29,30].

In this work, we propose an intelligent autonomous mobile delivery robot (MDR) using a deep learning approach. Our proposed method can recognize various traffic signs on different surfaces based on our dataset. The main contributions of this study are as follows:

1. We present the development of an autonomous mobile delivery robot that combines packages of delivery feature, a human-assistance feature, and an automatic movement feature. It uses a camera as the only sensor to detect images and deep learning techniques to recognize traffic line-markings.
2. Our autonomous mobile delivery robot can be driven across various surfaces, including a terrazzo floor, a canvas floor, and a wooden floor, in an environment with road markings similar to those in the training images, thus obviating the need for new road marking recognition technologies. We hypothesize that if a car were to be driven to many places across different road surfaces and in different environments but with the same traffic lanes, a robot would be able to move along designated directions without needing to be retrained regarding road marking recognition.
3. We combine the parcel-handling features of a factory assistant robot and the accessibility features of human-assistance robots to the proposed autonomous mobile delivery robot. These two features respectively provide the autonomous mobile delivery robot with the ability to transport packages and interact with humans. The developed robot does not only move automatically and deliver packages but can also move applicably in complicated office environments.

The remainder of this paper is organized as follows: The related works are reviewed in section two. The proposed system is presented in detail in section three. The experimental results and a discussion are given in section four. Finally, the conclusions section has been concluded in section five.

2 Related works

2.1 Autonomous robots

Autonomous robots are machines with the ability to move around in environments such as factories, houses, hospitals, apartments, and public places. Such robots can perceive objects in these environments and can make decisions. Generally, studies on robotic perception systems employ a camera sensor technology in conjunction with other sensors [31]. Sensors are critical devices that enable robots to receive input [32,33]. A camera is a sensor that can work with LiDAR, radar, and ultrasonic technologies. It can help in understanding the environment when provided with a stable infrastructure, such as traffic lane signs, speed signs, and speed limit signs [34, 35].

Regarding robot automatic movement features, researchers have examined the benefits and conducted studies related to road recognition. Typically, an autonomous robot was used to learn public and unstructured roads that do not have traffic lanes. The image processing technique has been used to recognize constructed and unconstructed roads with shadow and illumination changes and the neural network has been chosen to handle the lane recognition tasks. Qingji Gao and Qijun Luo SM [19] proposed the Rough Set Based Unstructured Road Detection (RSURD) method employed with the HSV color model to convert camera images to a color spectrum ranging from 0 to 256 levels. This method enables the robot to examine roads with complicated structures using a low-complexity algorithm. The histogram is used to build the color model of the roads and compare it in each pixel for each color model. However, it relied on tracking results from one image to another and were unable to recognize roads with different color properties. Kuo-Yu Chiu, *et. al.* [36] presented a color-based segmentation method for lane detection by selecting an area of interest (ROI), group the traffic lines by choosing a white line and applying a color threshold to separate the road and traffic colors, and finally use the line boundaries to detect the traffic line. This method can easily eliminate the influence due to sunlight, shadows on sidewalks, and obstacles such as vehicles and pedestrians. However, it depends on the manual settings of parameters. Zhang, F., *et. al.* [37] introduced a lane marking separation technique using Random Finite Set Statistics for estimating the position of road markings and using the Probability Hypothesis Density (PHD) filter. They match between the visual coordinate (u, v) of the tracked pixel and the corresponding point (x, y) on the ground plane (vehicle coordinate) to extract the characteristics of the traffic line. The PHD filters are used to reduce the computational difficulty of Bayes filters, resulting in higher cost on time complexity. Amaradi P. *et. al.* [38] presents the Hough transform technique of lane tracking and obstacle detection using LIDAR sensors to measure the distance drifted from the center of the lane to be able to detect obstacles. Kim ZW [39] presented the RANdom SAmple Consensus (RANSAC) algorithm to find the lane boundary hypothesis in real-time. The probability clustering algorithm is applied to group the lane boundary assumption as the left and right lane boundaries. They hypothesized left and right lane boundaries separately and identified the traffic lines with Intensity-Bump Detection, ANNs, NBCs, and SVM. The results show that SVMs show better performance than other classifiers, and ANNs are inferior in performance.

From the aforementioned works, our research is different as we do not consider the color of the surface where the robot moves. We use deep learning techniques to learn traffic lanes, allowing the robot to move on different surfaces. The proposed method is designed to allow the robot to move on various surfaces, including terrazzo, canvas, and wooden floors. The model has trained its movement from the dataset on a terrazzo floor environment and tested it on canvas and wooden floors.

2.2 Recognition using a convolutional neural network

The movement of the robot in the traffic lane is addressed in recognition using a convolutional neural network (CNN) to achieve end-to-end training. End-to-end platform [21, 22], a new model for self-driving cars (i.e., autonomous cars), was presented as a brain-inspired cognitive model with attention (CMA). This model functions to simulate the operation of the human brain. It describes the relationship between complex traffic scenes and the recurrent neural network updated using an attention mechanism and short-term memory. Eleftheriou G., *et. al.* [40] presents the development of a linear walking robot based on a Fuzzy-based PID closed-loop control system. The ruled-based method works in tandem with the computer vision system where system detects and recognizes the lines in front of the robot using a Pixy2 camera and reports them to the microcontroller board. These visual data enhance the PID controller's ability to respond on time to rapid nonlinear changes from preset values. Simmons B., *et al.* [41] proposed an end-to-end learning approach to help small remote-controlled cars run in indoor environments. The Deep Artificial Neural Network algorithm (CUA-DNN) and Convolution Neural Network Algorithm (CUA-CNN) were used in the training to map prototypes that presented the mechanical, electrical, and software design of self-driving cars. The accuracy and loss of the two neural networks were compared with VGG16 and DenseNet models. A finite state machine was used to control vehicle behavior when changing lanes and stopping state. Do TD., *et al.* [42] focused on finding a model that mapped the dataset to the predictive output of steering angle using deep neural networks. There are two parts; i) building a 1/10 scale RC car platform, computer, Raspberry Pi 3 Model B, and front camera; ii) is building a mock test road on a Raspberry Pi to autonomously drive in an outdoor environment around an oval and Figure 8 with a traffic sign. The results demonstrated the efficiency and strength of the automatic model in lane healing tasks.

3 Proposed methods

3.1 Data collection

We have collected a total of 120000 images, including 70000 forward, 24500 left turns, 24500 right turns, and 1000 stop sign images. We put the robot in the lane and operate it by hand control via a smartphone. The camera is set to capture 160×120 pixels with a frame rate of 20 fps. A camera was set with the Raspberry Pi, where it performs two functions: collecting image datasets for training and receiving image signals as input values for testing the model. While collecting the dataset, the microcontroller calculates the motion linear velocity and wheel angle for each image frame.

The main key factors to consider in this work are appearance and motion. The appearance is the presence of the traffic lanes and the stop signs in an image. We use the RGB and Grayscale image datasets with the same size (i.e., 160×120 pixels) to train the model. For training, the grayscale image model improves recognition performance [43, 44]. Therefore, we train the model with RGB and grayscale images to clarify which types of images are affected by the recognition performance of the model. While manipulating the robot to collect the images, the dataset is generated with one motion linear velocity value and one wheel angle value per image. The RGB color images are converted to grayscale images as shown in Eq (1):

$$Gray\ scale\ image = (W_1 * R) + (W_2 * G) + (W_3 * B), \tag{1}$$

where $W_1 + W_2 + W_3 = 1$ and W_1, W_2, and $W_3 > 0$. R, G, and B represent the red, green, and blue values at the conversion point, respectively, while W_1, W_2, and W_3 are the weights of each red, blue, and green, respectively.

During testing the robot's autonomous movement, it has been found that when the robot encounters an obstacle, it chooses to move on an unobstructed path. This allows the robot to be able to avoid obstacles.

For the motion, the feature extraction used in the movement of the robot includes linear velocity, wheel degrees, and motion characteristics. Motion linear velocity value and wheel angle value have occurred while collecting the image dataset, traffic lanes, and stop signs. These values are controlled manually via a smartphone. The motion linear velocity and the wheel degree are calculated by the program installed in the robot. The motion linear velocity is the velocity that occurs as the robot moves in a plane. As the robot moves through each point, a linear coordinate is obtained and represented in the form of Eq (2).

$$v_0 = mx + b, \tag{2}$$

where v_0 is the point at which the line crosses the y-intercept, x is the point where the line crosses the x-axis, m is the slope of the line, and b is the y-intercept.

The robot can move forward, turn left, or turn right. Forward motion is carried out in a straight line and can be calculated from the linear velocity as shown in Eq (3).

$$v = v_0 + at, \tag{3}$$

where v is the change in position from point to point, v_0 is the initial linear velocity, a is the acceleration, and t is time.

The wheel angle is the degree of wheel rotation caused by robot movement in the left and right directions as shown in Eq (4). The angle of the robot wheel can be changed to a nonlinear operation by input from the front axle to the nearest route point (cx, cy). The front-wheel steering position provides easy handling and helps the wheels to move correctly in the path without slipping off.

$$\delta(t) = \theta(t) + arctan\frac{kx(t)}{\upsilon_x(t)}, \tag{4}$$

where $\delta(t)$ is the degree of inclination of the robot's front wheels, $\theta(t)$ is the angle describing the trajectory of the robot, kx(t) is the increase in speed of the robot, and $\upsilon_x(t)$ is the speed of the robot over time *t*.

The center of the robot's front wheel from the nearest point along the edge, $\theta(t)$ is set to zero; $\upsilon_x(t)$ is the speed of the robot at time *t* and θ is the angle that describes the trajectory of the robot. Here, θ is the variable wheel angle, which can be calculated from Eq (5).

$$\theta = \frac{\omega}{t}, \tag{5}$$

where θ represents the wheel angle offset (in degrees), ω is the angular velocity, and *t* is time.

We have defined images as input data for the model and the motion linear velocity and wheel angle as output data. The motion linear velocity and wheel angle are used as a label for each image frame, leading to the movements of the robot as forward, left, right, and stop for each frame. The labels are represented as [1 0 0 0], [0 1 0 0], [0 0 1 0], and [0 0 0 1], denoting forward, left, right, and stop, respectively. The features of the robots are the image, motion linear velocity, and wheel angle. The motion linear velocity and wheel angle values were collected in the dataset. The advantage of performing on these features is that the robots perform well and work rapidly without undue complexity while yielding good results [45]. The motion and the wheel angle of the robot can be classified as forward, left, right with the criteria of $0<T<\eta$ for motion linear velocity and $0\leq T<\alpha$, $-\alpha<T<-1$, $\alpha<T<1$ for wheel angle, respectively. The stop class is classified with the criteria of $T=0$ for motion linear velocity and $0\leq T\leq 1$ for wheel angle. T represents the threshold, η denotes the motion linear velocity, with a starting value of 0, and α represents the wheel angle, with a starting value of 0.

3.2 Convolution Neural Network

Our proposed method is based on the CNN algorithm. We train a model to control the autonomous robot using the motion linear velocity, wheel degree, and motion characteristics as a label. Images and motion characteristics are defined as the input to predict linear velocity and wheel angle. The CNN structure is illustrated in Figure 1.

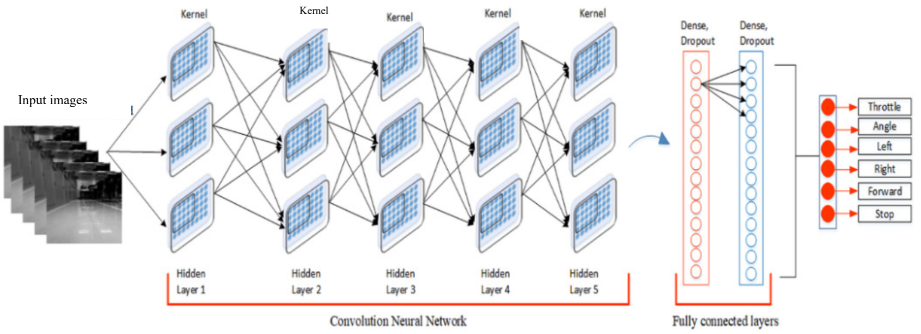


Fig. 1. Structure of the MDR. The CNN process in the hidden layers that contain the hyperparameters, including the Convolution Layer and Flattened Layer. In this study, using a dataset containing two-dimensional images, a 2D-CNN was used to train the models to recognize traffic lanes and stop signs

For training, we have prepared a data set of traffic lanes and stop signs images containing 120000 data, 96000 for training and 24000 for validation. The images of the traffic lanes and the stop signs were archived in the image array format and then used to train the model. We have designed a value hyperparameter for training models for forecasting motion characteristics and forecasting motion linear velocity and wheel angle. We chose to use a 2D convolution neural network (2D-CNN) because the input data is a 2D image [46]. The output of the model is forecasting motion linear velocity, wheel angle, motion characteristics, including forward, left, right, and stop. Our model is constructed by stacking five convolution layers and fully connected (FC) layers containing two dense layers with dropouts (rate = 0.1) [46, 48, 49]. Precisely, the first three convolution layers have a kernel size of 5 × 5 pixels and a stride s = 2 to reach a bridge representing the spatial data. The rest of convolution layers consist of a kernel size of 3 × 3 pixels with a stride s = 2. All convolution layers are followed by ReLU activation function [47]. The number of filters in each convolution layer is set from 24→32→64→64→64. The last convolution layer output is flattened before feeding it into FC layers. The number of neurons of FC layers is set from 100→50→6. Finally, we optimize the objective function of the model using ADAM optimizer [50] to achieve the optimal parameters of the model.

Training a CNN from image datasets to model an MDR is necessary to recognize and predict the direction of robot movement. The value of the dataset used in the training is represented as p and the function receiving the value of time is represented as t which is the actual value from the camera. Generally, MDR has learnable weights and biases to minimize annoying signals of inputs, causing the inputs to have more stability. The value of weight is represented as w. In case there is a weight mean in every period, the signal obtained from the sensor is more stable which can be written as the Eq 6.

$$s(t) = \int p(a)w(t-a)da, \tag{6}$$

where
$s(t)$ is position of the period,
$p(a)$ is function receiving the value of the dataset used in the training,
$w(t-a)$ is weight.

Receiving value from the camera sensor in every period is one time per second. t, p and w contain integer type. It can be written in a type of array as shown in the Eq 7.

$$s[t] = (p * w)(t) = \sum_{a=-\infty}^{\infty} p[a]\, w[t-a], \tag{7}$$

where
$s[t]$ is time function,
$p[a]$ is p function receiving value of the dataset,
$w[t-a]$ is w function is represented by the kernel.

The input and the kernel is a multidimensional array that can be used as parameters for the model training. A multidimensional array is called tensors. Each element of input and kernel is stored separately. Since the image dataset is two-dimensional input data, and the kernel is determined to have two dimensions of size 5x5 which can be written as Eq 8.

$$s[i,j] = (I * K)[i,j] = \sum_m \sum_n I[m,n] K[i-m, j-n], \tag{8}$$

where
$s[i,j]$ is variables collecting output values,
I is a dataset for the training,
K is a kernel,
$(I * K)[i,j]$ is Convolution implementation on 2D dataset and kernel in the training,
$\sum_{\mathrm{n}} \sum_n I[m,n]\, K[i-m, j-n]$ is the total amount of convolution implementation and 2D kernels.

4 Experimental results and discussion

4.1 Evaluation criteria

We evaluated a predicted motion characteristics model based on accuracy, precision, and recall. Accuracy is an efficacy indicator that measures the accuracy and precision of a model. The model achieves a predictive level close to the actual robot movement direction. The formula of Precision and recall are calculated as $Recall = \frac{TP}{TP+FN}, Presion = \frac{TP}{TP+FP}$. The formula of accuracy is $Accuracy = \frac{TP+TN}{N}$, where *TP* represents a true positive, *TN* denotes true negatives, *FN* is a false negative, *FP* is a false positive, and *N* is the total number of datasets.

4.2 Implementation details

We use GPU for processing the model. We chose the value of motion velocity as 0.8 as it provides the best performance in controlling the robot. The dataset for the model training was 80% and validation was 20%. The laboratory is set with three different scenes, including the terrazzo floor, canvas, and wooden floor. The side traffic lanes were bounded by solid white lines, and a yellow dashed line marked the middle of the road. The floor in the laboratory was bright white.

4.3 Performance evaluation

The experimental results of robot movement direction using RGB and grayscale input images are shown in Tables 1 and 2, respectively. We present the accuracy for both RGB and grayscale input images in Table 3.

Table 1. Evaluation of robot movement direction with RGB images input data

Class	Precision	Recall
1 (Left)	0.89	0.83
2 (Right)	0.84	0.82
3 (Forward)	0.91	0.94
4 (Stop)	0.86	0.98

From Table 1, we have used the input data as RGB images and found that the highest precision and recall of the robot's movement is the forward direction with 91.00% and 94.00%, respectively.

Table 2. Evaluation of robot movement direction with grayscale images input data

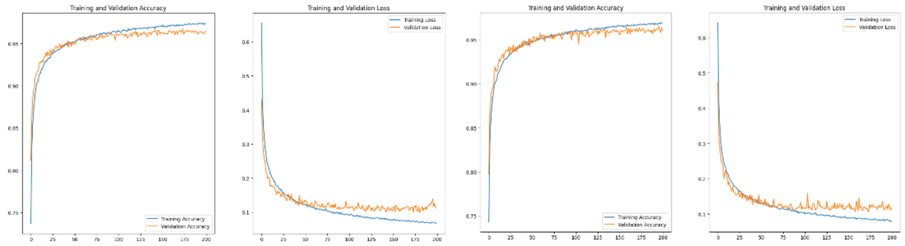
Class	Precision	Recall
1 (Left)	0.87	0.83
2 (Right)	0.88	0.81
3 (Forward)	0.91	0.95
4 (Stop)	0.92	1.00

From Table 2, we have used the grayscale images as input data and found that the stop class has the highest precision at 92.00% and recall at 100%. This shows that the model can predict the stop motion most accurately, making it possible to test the movement of the robot in the real environment where the robot stops moving when it reaches the stop sign.

Table 3. Evaluation of robot movement direction prediction

Category	Training Accuracy	Validation Accuracy
RGB images input data	96.92%	96.01%
Grayscale images input data	97.36%	96.45%

Table 3 shows the forecast training and validation accuracy of the model in comparison of the input data images between RGB and grayscale images to test how the type of image affects the accuracy of the model. It is clearly shown that grayscale images influence the accuracy of the model for both training and validation than using RGB images. The training and validation accuracy and loss of RGB images and grayscale images are illustrated in Figures 2.



(a) (b) (c) (d)

Fig. 2. Model training and model testing using the RGB and grayscale dataset. (a) showing the accuracy of model training and testing of RGB dataset, (b) showing model training and testing losses of RGB dataset, (c) showing the accuracy of model training and testing of grayscale dataset and (d) showing model training and testing losses of grayscale dataset

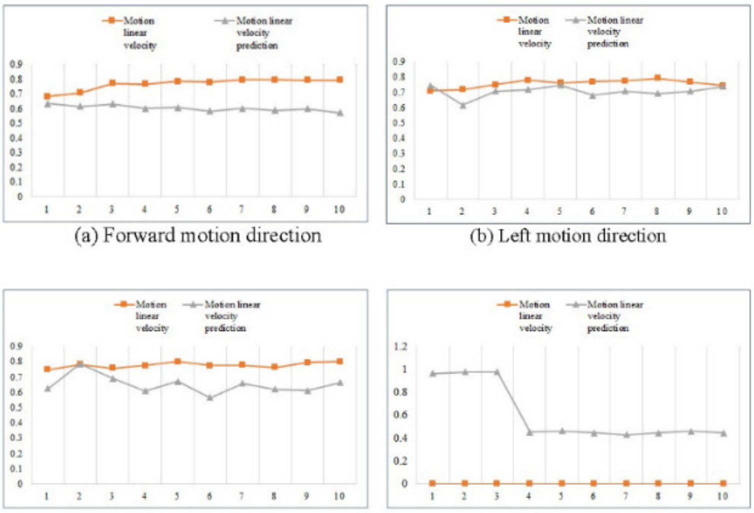
From Figure 2, the validation values of RGB training models are about 0.95 and grayscale models are slightly greater than 0.95. From Table 3, the training accuracy of the grayscale image model is 97.36%, and the validation accuracy of 96.45%, both of which are higher than models that use RGB images. As a result, the training model using grayscale images had good recognition performance. For training and validation loss values, it was seen that the model using grayscale image had a validation loss slightly lesser then using RGB image, meaning that this model accurately predicts the direction of the robot's movement as the validation loss approaches zero.

For forecasting motion linear velocity and wheel angle, we evaluated the model's performance using the relationship between the *x* and *y* variables. During the prediction of the robot movement direction using the MDR technique, the traffic lane and stop sign image dataset are used in the data for variable X to create a model for robot traffic lane recognition. The motion linear velocity, wheel angle, and class datasets for the Y variable are used to predict the direction of robot motion. The tests were conducted ten times to obtain the average motion linear velocity shown in Figure 3 and the wheel angles shown in Figure 4 in order to determine the robot movement direction while reading the traffic lines in real-time during robot motion. Since the motion linear velocity and wheel angle are numerical values, we used the Root Mean Square Error Method (RMSE) calculated as $RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$, where Σ is a summation, y_i is the dataset, $\hat{y}_i$ denotes the forecast values, and n is the total amount of data used per calculation. The evaluation results are shown in Table 4.

Table 4. RMSE Evaluation of the robot movement direction prediction

Category	**RGB Images (RMSE)**		**Grayscale Image (RMSE)**	
	Motion linear velocity	***Wheel angle***	***Motion linear velocity***	***Wheel angle***
Forward	0.0069	0.0048	0.0043	0.0047
Left	0.0044	0.0081	0.0034	0.0077
Right	0.0104	0.0092	0.0089	0.0089
Stop	0.0013	0.0465	0.0004	0.0198
Average	0.0057	0.0171	0.0042	0.0102

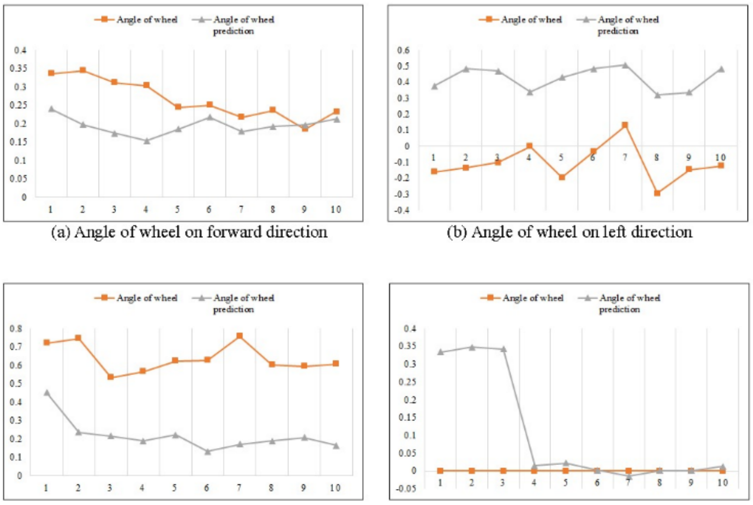
From Table 4, we assessed the performance for model discrepancies. We assigned the input data to RGB and grayscale images to predict motion linear velocity and wheel angle. Our experiments revealed that grayscale images resulted in the model in fewer dynamics based on the mean of wheel angle, which is the value that determines the direction of movement for the robot. The error is closest to 0 (i.e., 0.0102). Considering at each motion characteristic, the grayscale input data results in the model predicting more accurately for forward, left, right, and stop motion than using the RGB image input data. From the forecast data of the movement characteristics of the robot, we compared the values of motion linear velocity and wheel angel with the motion linear velocity and wheel angel data obtained from the data collection section as shown in Figures 3 and 4. From Figure 3, the curve with the square symbols (motion linear velocity) corresponds to the motion linear velocity data. The curve with the triangle symbols shows the motion linear velocity prediction of the model. The graph shows that the values forecasted by the models are closely related to the ground-truth values in the dataset. From Figure 4, the curve with square symbols (wheel angle) shows the wheel angle values in the dataset, which are used to determine the motion direction of the robot. The curve with triangle symbols (predicted wheel angle) shows the wheel angles predicted by the model. Therefore, the values forecasted by the models indicate wheel angles in the same direction as the ground-truth values in the dataset.



(a) Forward motion direction (b) Left motion direction

(c) Right motion direction (d) Stop motion

Fig. 3. Forecast of the motion linear velocity in each motion direction



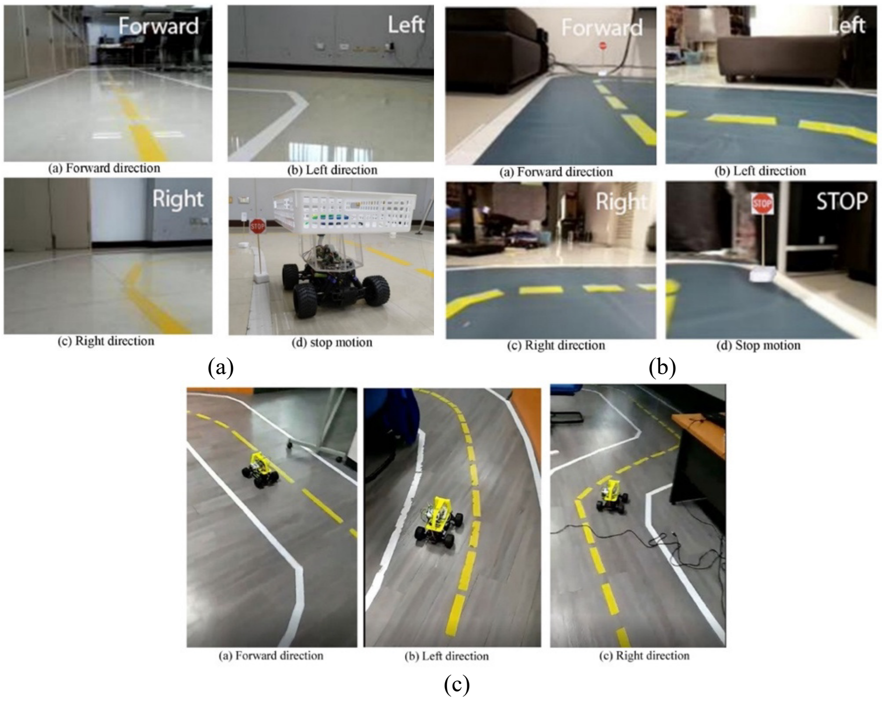
(a) Angle of wheel on forward direction (b) Angle of wheel on left direction

(c) Angle of wheel on right direction (d) Angle of wheel on stop motion

Fig. 4. Forecast of the wheel angle in each motion direction

4.4 Experimental results

We test the movement of the robot with three different surfaces considering various actions, e.g., turning left, turning right, going forward, and stopping, as shown in Figure 5.



(a) Forward direction (b) Left direction (c) Right direction (d) stop motion

(a)

(a) Forward direction (b) Left direction (c) Right direction (d) Stop motion

(b)

(a) Forward direction (b) Left direction (c) Right direction

(c)

Fig. 5. Movement of the robot across (a) a terrazzo floor, (b) a canvas or sailcloth, and (c) a wooden floor

4.5 Model comparison

We experiment with the comparison of the forecast motion characteristics. The results are shown in Table 5.

Table 5. Comparison of the evaluation of robot movement direction prediction

Model	Training accuracy	Validation accuracy	Parameters
CUA-DNN (41)	84.36%	84.31%	89,732
CUA-CNN (41)	97.28%	94.61%	1,050,500
VGG16 (41)	91.80%	86.32%	46,112,540
CNN (42)	92.38%	89.04%	267,291
MDR + RGB images (proposed method)	96.92%	96.01%	266,730
MDR + Grayscale images (proposed method)	97.36%	96.45%	266,730

Table 5 shows the model's performance evaluation on motion characteristics: forward, left, right, and stop. We compared our proposed algorithm with the other state-of-the-art works. It was found that our proposed method had the best validation accuracy of MDR-grayscale at 96.45%. Thus, grayscale images affect the forecasting performance of the model as our hypothesis. Besides, grayscale images affect not only the predictive performance of the model but also the recognition of the model, with the model having a recognition efficiency of 97.36%.

For forecasting motion linear velocity and wheel angle, we use the error measurement tool, RMSE, due to the consecutively numerical values of prediction output. We experimented with four scenarios. The results obtained from the model performance evaluation are shown in Table 6.

Table 6. Comparison of the evaluation of robot movement direction prediction with feature extraction

Model	**RMSE**	
	Motion velocity	***Wheel angle***
MDR + Grayscale images + motion characteristics (proposed)	0.0032	0.0025
MDR + RGB images + motion characteristics (proposed)	0.0033	0.0033
MDR + Grayscale images (proposed)	0.0032	0.0043
MDR + RGB images (proposed)	0.0037	0.0056

Table 6 shows an RMSE comparison of model performance for forecasting motion linear velocity and wheel angle. We conducted model training with two groups of experiments: (1) we trained the model with RGB image input data and grayscale image, and (2) we trained the model with RGB image input data, grayscale image, and motion characteristic data (forward, left, right, and stop). We hypothesized that grayscale images affect the model's performance in forecasting. It was seen that training the model with the grayscale image input data (MDR + Grayscale images) had inaccuracies in predicting model performance less than training the RGB image input data model (MDR + RGB images) because its RMSE is closer to 0, satisfying the hypothesis. We also hypothesized that if we input motion characteristics to the model, it will optimize the model, allowing the model to better predict the direction of movement of the robot. From the experiments shown in Table 5, the deviations in the direction of motion forecasting of the models trained with the Grayscale images dataset and motion characteristics (MDR + Grayscale images + motion characteristics) were less than the models trained with the RGB images and motion characteristics (MDR + RGB images + motion characteristics). Thus, from the experiments, grayscale images affect the recognition and predictive performance of the model. This makes the model more recognizable and predicts the direction of the robot's movement with less error. In addition, the application of motion characteristics as input to the model allows the model to predict the direction of motion more accurately.

5 Conclusions

We proposed a novel study of an intelligent autonomous document mobile delivery robot equipped with deep learning in which the recognition system uses a camera sensor. During the model training process, a dataset was collected from practice in a laboratory, and a deep learning technique was trained to perform traffic lane recognition. After training, the trained model was tested by a robot and evaluated its movement within the specified lanes in the laboratory. It was hypothesized that if the model that was obtained from the training were to be used in other places with lanes similar to those in the laboratory but with a different floor color, the model would still be able to control the robot moving direction in the same way as in the laboratory. Therefore, a simulation model of lanes was built on an opaque gray canvas to test the movement of the robot. The model trained with the MDR technique was tested by allowing the robot to move automatically across a terrazzo floor in the laboratory and across a canvas laid out on the floor in the laboratory for the simulation. The results showed that the robot can move to the left, to the right, forward, and stop correctly. The efficacy of the robot was evaluated based on precision, recall, and accuracy metrics. The evaluation was performed for each class: forward, left, right, and stop. The mean precision obtained from the evaluation results of all four classes showed that the model provides correct predictions of 88.91%. The recall reveals how likely the model is to respond correctly. The result obtained from the prediction reflects its probability of being correct. The experiments indicated that the mean recall and the mean accuracy of this model are 90.49% and 96.31%, respectively. This work has several strengths. First, the proposed robot works effectively with small requirement of the equipment. The advantages are that our robot is small, low cost, and able to work in real situations. Besides, the proposed method is efficient and flexible for real world scenes as it can be applied to various locations with different floor colors and types. Specifically, when there is a change in moving direction, new lanes do not need to be constructed; only the positions of the stop signs need to be changed. For future work, we plan to add sensors to the robot recognition system, enabling the robot to intelligently recognize environments and create a Simultaneous Localization and Mapping (SLAM). It can be included a robot operating system, allowing the robot to move in the designated directions more precisely and accurately.

6 Acknowledgments

This work was supported by the Department of Mathematics, King Mongkut's University of Technology Thonburi and Thailand Science Research and Innovation Fund, King Mongkut's University of Technology North Bangkok with Contract no. KMUT-NBBasicR-64-03.

7 References

[1] L. Sabattini et al., "The PAN-Robots Project: Advanced Automated Guided Vehicle Systems for Industrial Logistics," in IEEE Robotics & Automation Magazine, vol. 25, no. 1, pp. 55-64, March 2018. https://doi.org/10.1109/MRA.2017.2700325

[2] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating Hundreds of Cooperative, Autonomous Vehicles in Warehouses", *AIMag*, vol. 29, no. 1, p. 9, Mar. 2008.

[3] H. Andreasson et al., "Autonomous Transport Vehicles: Where We Are and What Is Missing," in IEEE Robotics & Automation Magazine, vol. 22, no. 1, pp. 64-75, March 2015. https://doi.org/10.1109/MRA.2014.2381357

[4] G. Garibotto, S. Masciangelo, M. Ilic and P. Bassino, "Service robotics in logistic automation: ROBOLIFT: vision based autonomous navigation of a conventional fork-lift for pallet handling," 1997 8th International Conference on Advanced Robotics. Proceedings. ICAR'97, 1997, pp. 781-786.

[5] G. Garibotto, S. Masciangelo, P. Bassino, C. Coelho, A. Pavan and M. Marson, "Industrial exploitation of computer vision in logistic automation: autonomous control of an intelligent forklift truck," Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146), vol. 2, 1998, pp. 1459-1464.

[6] L. Vladareanu, "Robot Extenics Control Developed by Versatile, Intelligent and Portable Robot Vipro Platform Applied on Firefighting Robots", Int. J. Onl. Eng., vol. 16, no. 08, pp. pp. 99–107, Jul. 2020. https://doi.org/10.3991/ijoe.v16i08.15309

[7] N. Ramdani et al., "A Safe, Efficient and Integrated Indoor Robotic Fleet for Logistic Applications in Healthcare and Commercial Spaces: The ENDORSE Concept," 2019 20th IEEE International Conference on Mobile Data Management (MDM), 2019, pp. 425-430. https://doi.org/10.1109/MDM.2019.000-8

[8] H. Pikner, R. Sell, K. Karjust, E. Malayjerdi and T. Velsker, "Cyber-physical Control System for Autonomous Logistic Robot," 2021 IEEE 19th International Power Electronics and Motion Control Conference, 2021, pp. 699-704. https://doi.org/10.1109/PEMC48073.2021.9432526

[9] L.M. Duan, "Path Planning for Batch Picking of Warehousing and Logistics Robots Based on Modified A* Algorithm", *Int. J. Onl. Eng.*, vol. 14, no. 11, pp. pp. 176–192, Nov. 2018. https://doi.org/10.3991/ijoe.v14i11.9527

[10] B. Kehoe, S. Patil, P. Abbeel and K. Goldberg, "A Survey of Research on Cloud Robotics and Automation," in IEEE Transactions on Automation Science and Engineering, vol. 12, no. 2, pp. 398-409, April 2015. https://doi.org/10.1109/TASE.2014.2376492

[11] E. Cardarelli, V. Digani, L. Sabattini, C. Secchi, C. Fantuzzi, "Cooperative cloud robotics architecture for the coordination of multi-AGV systems in industrial warehouses," Mechatronics, vol. 45, pp. 1–13. 2017. https://doi.org/10.1016/j.mechatronics.2017.04.005

[12] Y. Dobrev, M. Vossiek, M. Christmann, I. Bilous and P. Gulden, "Steady Delivery: Wireless Local Positioning Systems for Tracking and Autonomous Navigation of Transport Vehicles and Mobile Robots," in IEEE Microwave Magazine, vol. 18, no. 6, pp. 26-37, Sept.-Oct. 2017. https://doi.org/10.1109/MMM.2017.2711941

[13] A. Ewald and V. Willhoeft, "Laser scanners for obstacle detection in automotive applications," Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No.00TH8511), 2000, pp. 682-687.

[14] Y. Zhou, H. Hu, Y. Liu and Z. Ding, "Collision and Deadlock Avoidance in Multirobot Systems: A Distributed Approach," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 7, pp. 1712-1726, July 2017. https://doi.org/10.1109/TSMC.2017.2670643

[15] F. Oleari, M. Magnani, D. Ronzoni and L. Sabattini, "Industrial AGVs: Toward a pervasive diffusion in modern factory warehouses," 2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing, 2014, pp. 233-238. https://doi.org/10.1109/ICCP.2014.6937002

[16] S. Silberstein, D. Levi, V. Kogan and R. Gazit, "Vision-based pedestrian detection for rear-view cameras," 2014 IEEE Intelligent Vehicles Symposium Proceedings, 2014, pp. 853-860. https://doi.org/10.1109/IVS.2014.6856399

[17] R. Benenson, M. Omran, J. Hosang, B. Schiele, "Ten Years of Pedestrian Detection, What Have We Learned?," in Agapito L, Bronstein MM, Rother C, (eds) Computer Vision - ECCV 2014 Workshops, Springer, pp. 613–27, 2015. https://doi.org/10.1007/978-3-319-16181-5_47

[18] M. Luo, "Matlab-Realized Visual A* Path Planning Teaching Platform", *Int. J. Emerg. Technol. Learn.*, vol. 13, no. 10, pp. pp. 196–207, Oct. 2018. https://doi.org/10.3991/ijet.v13i10.9463

[19] Q. Gao, Q. Luo and S. Moli, "Rough Set based Unstructured Road Detection through Feature Learning," 2007 IEEE International Conference on Automation and Logistics, 2007, pp. 101-106. https://doi.org/10.1109/ICAL.2007.4338538

[20] Y. Zhang, Q. Li, H. Lu, X. Liu, X. Huang, C. Song, "Optimized 3D street scene reconstruction from driving recorder images," in Remote Sens, vol. 7, no. 7, pp. 9091-9121, 2015. https://doi.org/10.3390/rs70709091

[21] P. Viswanath, S. Nagori, M. Mody, M. Mathew and P. Swami, "End to End Learning based Self-Driving using JacintoNet," 2018 IEEE 8th International Conference on Consumer Electronics - Berlin, 2018, pp. 1-4. https://doi.org/10.1109/ICCE-Berlin.2018.8576190

[22] S. Chen, S. Zhang, J. Shang, B. Chen and N. Zheng, "Brain-Inspired Cognitive Model With Attention for Self-Driving Cars," in IEEE Transactions on Cognitive and Developmental Systems, vol. 11, no. 1, pp. 13-25, March 2019. https://doi.org/10.1109/TCDS.2017.2717451

[23] A. Di Fava, M. Satler and P. Tripicchio, "Visual navigation of mobile robots for autonomous patrolling of indoor and outdoor areas," 2015 23rd Mediterranean Conference on Control and Automation, 2015, pp. 667-674. https://doi.org/10.1109/MED.2015.7158823

[24] Z. Chen and X. Huang, "End-to-end learning for lane keeping of self-driving cars," 2017 IEEE Intelligent Vehicles Symposium, 2017, pp. 1856-1860. https://doi.org/10.1109/IVS.2017.7995975

[25] Y. Choe and M. J. Chung, "System and software architecture for autonomous surveillance robots in urban environments," 2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence, 2012, pp. 535-536. https://doi.org/10.1109/URAI.2012.6463065

[26] A. M. Pinto, P. G. Costa and A. P. Moreira, "An architecture for visual motion perception of a surveillance-based autonomous robot," 2014 IEEE International Conference on Autonomous Robot Systems and Competitions, 2014, pp. 205-211. https://doi.org/10.1109/ICARSC.2014.6849787

[27] H. Guo, S. Cai, X. Wu, Q. Wu, W. Feng and Q. Gao, "Running person detection from a community patrol robot," 2016 IEEE International Conference on Information and Automation, 2016, pp. 1521-1525. https://doi.org/10.1109/ICInfA.2016.7832060

[28] S. Hoshino and S. Ugajin, "Adaptive patrolling by mobile robot for changing visitor trends," 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 104-110. https://doi.org/10.1109/IROS.2016.7759041

[29] D. Drake, S. Koziol and E. Chabot, "Mobile Robot Path Planning With a Moving Goal," in IEEE Access, vol. 6, pp. 12800-12814, 2018. https://doi.org/10.1109/ACCESS.2018.2797070

[30] C. Li, F. Wang, L. Zhao, Y. Li, Y. Song, “An improved chaotic motion path planner for autonomous mobile robots based on a logistic map,” International Journal of Advanced Robotic Systems, vol. 10, no. 6, 2013, p.273. https://doi.org/10.5772/56587

[31] H. Moravec, “Rise of the Robots,” Scientific American, vol. 281, no. 6, 1999, pp.124–35. https://doi.org/10.1038/scientificamerican1299-124

[32] N. Evans, “Automated Vehicle Detection and Classification using Acoustic and Seismic Signals,” Doctoral dissertation, University of York, 2010.

[33] S. Bapat, V. Kulathumani and A. Arora, "Reliable estimation of influence fields for classification and tracking in unreliable sensor networks," 24th IEEE Symposium on Reliable Distributed Systems, 2005, pp. 60-69.

[34] DH. Walling, AL. Wicks, SC. Southward, AT. Asbeck, “The Design of an Autonomous Vehicle Research Platform,” Doctoral dissertation, Virginia Tech, 2013.

[35] D.J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," Sensors, vol. 21, no. 6, 2021, p.2140. https://doi.org/10.3390/s21062140

[36] Kuo-Yu Chiu and Sheng-Fuu Lin, "Lane detection using color-based segmentation," IEEE Proceedings. Intelligent Vehicles Symposium, 2005, pp. 706-711. https://doi.org/10.1109/IVS.2005.1505186

[37] F. Zhang, H. Stähle, C. Chen, C. Buckl and A. Knoll, "A lane marking extraction approach based on Random Finite Set Statistics," 2013 IEEE Intelligent Vehicles Symposium, 2013, pp. 1143-1148. https://doi.org/10.1109/IVS.2013.6629620

[38] P. Amaradi, N. Sriramoju, Li Dang, G. S. Tewolde and J. Kwon, "Lane following and obstacle detection techniques in autonomous driving vehicles," 2016 IEEE International Conference on Electro Information Technology, 2016, pp. 0674-0679. https://doi.org/10.1109/EIT.2016.7535320

[39] Z. Kim, "Robust Lane Detection and Tracking in Challenging Scenarios," in IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 1, pp. 16-26, March 2008. https://doi.org/10.1109/TITS.2007.908582

[40] G. Eleftheriou, L. Doitsidis, Z. Zinonos and S. A. Chatzichristofis, "A Fuzzy Rule-Based Control System for Fast Line-Following Robots," 2020 16th International Conference on Distributed Computing in Sensor Systems, 2020, pp. 388-395. https://doi.org/10.1109/DCOSS49796.2020.00068

[41] B. Simmons, P. Adwani, H. Pham, Y. Alhuthaifi and A. Wolek, "Training a Remote-Control Car to Autonomously Lane-Follow using End-to-End Neural Networks," 2019 53rd Annual Conference on Information Sciences and Systems, 2019, pp. 1-6. https://doi.org/10.1109/CISS.2019.8692851

[42] T. -D. Do, M. -T. Duong, Q. -V. Dang and M. -H. Le, "Real-Time Self-Driving Car Navigation Using Deep Neural Network," 2018 4th International Conference on Green Technology and Sustainable Development, 2018, pp. 7-12.

[43] H. M. Bui, M. Lech, E. Cheng, K. Neville and I. S. Burnett, "Using grayscale images for object recognition with convolutional-recursive neural network," 2016 IEEE Sixth International Conference on Communications and Electronics, 2016, pp. 321-325. https://doi.org/10.1109/CCE.2016.7562656

[44] C. Kanan and G.W. Cottrell, “Color-to-grayscale: Does the method matter in image recognition?,” PLoS One, vol. 7, no. 1, 2012. https://doi.org/10.1371/journal.pone.0029740

[45] E. Gilbert and D. Johnson, "Distance functions and their application to robot path planning in the presence of obstacles," in IEEE Journal on Robotics and Automation, vol. 1, no. 1, pp. 21-30, March 1985. https://doi.org/10.1109/JRA.1985.1087003

[46] R. Yamashita, M. Nishio, R.K.G. Do, K. Togashi, “Convolutional neural networks: an overview and application in radiology,” Insights Imaging, vol. 9, no. 4, 2018, pp. 611–29. https://doi.org/10.1007/s13244-018-0639-9

[47] X. Glorot, A. Bordes, B. Yoshua, “Deep Sparse Rectifier Neural Networks,” In Proceedings of the fourteenth international conference on artificial intelligence and statistics, pp. 315-323, 2011.

[48] E. Bochinski, T. Senst and T. Sikora, "Hyper-parameter optimization for convolutional neural network committees based on evolutionary algorithms," 2017 IEEE International Conference on Image Processing, 2017, pp. 3924-3928. https://doi.org/10.1109/ICIP.2017.8297018

[49] Y. Ozaki, M. Yano, M. Onishi, “Effective hyperparameter optimization using Nelder-Mead method in deep learning,” PSJ Transactions on Computer Vision and Applications vol. 9, no. 1, 2017, pp.1-12. https://doi.org/10.1186/s41074-017-0030-7

[50] D.P. Kingma, J.L. Ba, “Adam: A method for stochastic optimization,” 3rd Int Conf Learn Represent ICLR 2015 - Conf Track Proc. 2015, pp. 1–15.

8 Authors

Thittaporn Ganokratanaa is University Lecturer in Applied Computer Science Programme, King Mongkut’s University of Technology Thonburi, Bangkok, Thailand. She is the IEEE Region 10 Student Activities Committee and IEEE MGA SAC Awards Committee. Her research of interests are anomaly detection, image and video analysis, and human-computer interaction (email: thittaporn.gan@kmutt.ac.th).

Mahasak Ketcham received the B.B.A. in Business Computer Siam University, Thailand, MS.I.Ed. Computer Technology, King Mongkut’s University of Technology North Bangkok, Thailand and Ph.D. in Computer Engineering, Chulalongkorn University, Thailand. He is an assistant professor in Department of Information Technology Management, King Mongkut’s University of Technology North Bangkok, Thailand (email: mahasak.k@itd.kmutnb.ac.th).

Article submitted 2022-04-28. Resubmitted 2022-09-06. Final acceptance 2022-09-06. Final version published as submitted by the authors.