



CNN-Based Self Localization Using Visual Modelling of a Gyrocompass Line Mark and Omni-Vision Image for a Wheeled Soccer Robot Application

Rudy Dikairono¹ Setiawardhana¹ Djoko Purwanto^{1*} Tri Arief Sardjono²

¹*Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia*

²*Department of Biomedical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia*

* Corresponding author's Email: djoko@ee.its.ac.id

Abstract: The Convolutional Neural Network (CNN) is an object classification method that has been widely used in recent research. In this paper, we propose CNN for use in the self-localization of wheeled soccer robots on a soccer field. If the soccer field is divided into equally sized quadrants with imaginary vertical and horizontal lines intersecting in the middle of the field, then the soccer field has an identical shape for each quadrant. Every quadrant is a reflection of the other quadrants. Superficially similar images appearing in different positions may result in positioning mistakes. This paper proposes a solution to this problem by using a visual modelling of the gyrocompass line mark and omni-vision image for the CNN-based self-localization system. A gyrocompass is used to obtain the angle of the robot on the soccer field. A 360° omni-vision camera is used to capture images that cover all parts of the soccer field wherever the robot is located. The angle of the robot is added to the omni-vision image using the visual modelling method. The implementation of self-localization without visual modelling gives accuracy rates of 0.3262, and this result is increased to 0.6827 with the proposed methods. The experiment was carried out in the robotics laboratory of the Institut Teknologi Sepuluh Nopember (ITS) with the ITS Robot with Intelligent System (IRIS) robot.

Keywords: Convolutional neural network, Self-localization, Visual modelling, Gyrocompass line mark, Wheeled soccer robot.

1. Introduction

One popular application of mobile robot is a soccer robot since soccer game is a very popular sport that attracts a wide audience around the world. The popularity of the sport translates to great academic and research interests, evidenced by soccer robot competitions that have been held since 1997 [1]. There are two competitions in RoboCup for wheeled soccer robot, which are Small-Size League (SSL) and Middle-Size League (MSL). Research on ball position for MSL Goal Keeper robot has been discussed in [2]. Studies on motion controls of SSL soccer robot with four wheeled omnidirectional has been conducted in [3, 4]. Modeling of three-wheels omnidirectional of MSL soccer robot has been discussed in [5]. In addition, the key element of a mobile robot to be successfully operate is the navigation such as localization and collision

avoidance. A method of collision avoidance based on velocity control has been proposed in [6]. Localization in outdoor environment often employs Global Positioning Systems (GPS) such as mobile robot for gas level mapping [7]. A web-camera to replace GPS is employed in wheel chair as a mobile robot for outdoor localization [8]. The study employs and compares two methods: using a single Convolutional Neural Network (CNN) and using Faster Regional-CNN for landmark detection trained using feedforward neural network (FFNN). Yet, the implementation will be more difficult and challenging for soccer field localization with very few features, containing only the field lines as a permanent feature.

Most current soccer robots use compass, odometry and camera for navigation as they are required to do accurate and fast localization. The use of omnidirectional camera can ease and improve the

performance of the localization, such as work in [9] that used CNN and proved that CNN was the best for feature extraction in omnidirectional images, the implementation utilized two-wheeled differential steering and topological maps. Using the same robot, localization approach with reject option is also proposed in [10] based on topological maps using Nearest Neighbors classifier and Local Binary Patterns feature extraction. In [11], CNN using omni-vision for MSL soccer robot is discussed using two stages of approach: opponent's robot detection and robot localization using the depth point cloud. The use of Kinect for depth sensor will increase the burden of CPU, and the CNN here is not used for direct self-localization. CNN framework for obstacle avoidance using a monocular webcam is also studied in [12]. For Human Object Identification for Human-Robot Interaction, Fast R-CNN is used in [13], and hybrid Fuzzy-CNN in [14]. Besides, the implementation of CNN has been used for many applications, such as for matching near duplicate image [15], for situation prediction and sentiment analysis [16, 17], for handwritten recognition [18], and for human brain segmentation in medical [19].

This paper will focus on implementing CNN in determining the position of MSL soccer robots in the field by using a visual modelling of the gyrocompass line mark and omni-vision image to do self-localization. This method is expected to result in better accuracy without sacrificing data processing speed. Previous study of self-localization of MSL soccer robot using omnidirectional camera also studied in [20] by generating the searching space based on white line information of soccer field by optimizing the fitness function using genetic algorithm (GA). However, GA will cost time of processing. Method to obtain X-junction, T-junction and L-junction that can help robot localization is also discussed in [21]. This method also cannot have the whole picture of the field and may be blocked by robots, while CNN can see overall features on the field not only the junctions.

Although we can use the field lines as a feature to obtain the player's position in the soccer field, the field lines themselves are a mirror of the x and y axes that are drawn perpendicularly from the center of the soccer field. The shape of the lines in each quadrant is a reflection of the other quadrants. This makes it difficult for AI systems to recognize the position of robots in the field only by using the field line features. Additional information on the direction of the field for the robot can be used to distinguish each quadrant from its mirror. A gyrocompass is used to obtain information on the robot's direction relative to the soccer field.

The proposed method uses a special marker to improve the localization results. The contributions of this paper are as follows:

- Proposing a new method of self-localization for wheeled soccer robots using CNN with Omni-vision image data coupled with a gyrocompass line mark. The addition of visual models of compass data can significantly improve the accuracy of the CNN system.
- Proposing an original data set in the form of an Omni-vision image and gyrocompass line mark of the robot in the soccer field is presented. This data set makes reference to the robot's position using a global camera on the soccer field. Each image makes reference to the robot's position and direction, which is presented in the file name of the images.

The next part of this paper is divided as follows: Section 2 discusses the design and implementation of the wheeled soccer robots that have been developed by the IRIS team. Section 3 presents the proposed method in the form of the self-localization of wheeled soccer robots. The experiments for the data set collection and the processing technique are presented in Section 4. Furthermore, Section 4 also contains the results of the experiments that were carried out in both simulations and directly on the robot, which are then summarized in the conclusion in Section 5.

2. Wheeled soccer robot

This paper uses the IRIS robot to obtain the data

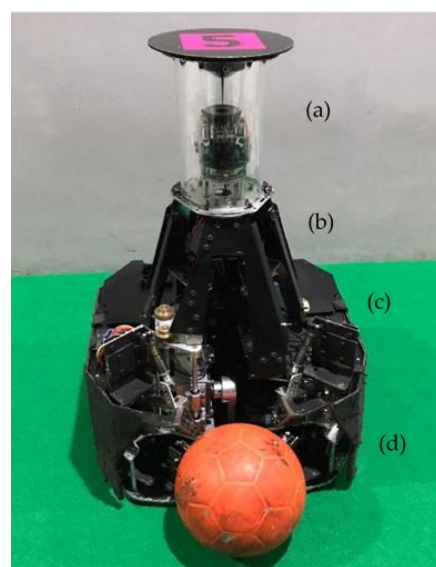


Figure. 1 ITS's robot with Intelligent system (IRIS) wheeled soccer robot platform: (a) omni-vision camera, (b) computer and microcontroller, (c) gyrocompass sensor, and (d) ball shooter, ball catcher, and omni-wheels

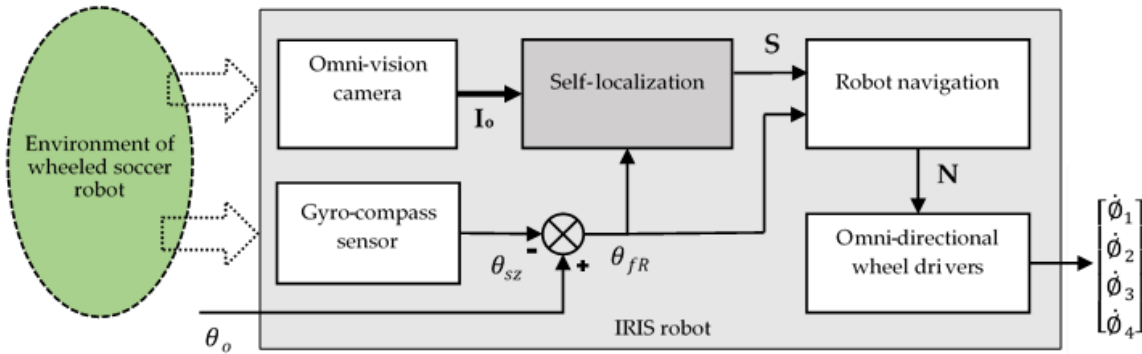


Figure. 2 Functional diagram of the IRIS robot. I_o is the omni-vision image; θ_o is the offset angle of the IRIS robot coordinate system to the soccer field coordinate system; θ_{sz} is the output angle of the gyro compass sensor; θ_{fR} is the angle of the IRIS robot coordinate system on the soccer field coordinate system; S is the matrix output of self-localization; N is the matrix output of the robot navigation; $\dot{\phi}_1$ is the angular speed of wheel 1; $\dot{\phi}_2$ is the angular speed of wheel 2; $\dot{\phi}_3$ is the angular speed of wheel 3; and $\dot{\phi}_4$ is the angular speed of wheel 4

set and implementation of the proposed method. The IRIS robot is a wheeled soccer robot, as seen in Fig. 1. The robot consists of an omni-vision camera, computer system, microcontroller system, gyrocompass sensor, and omni-wheel system. An omni-vision camera is used to capture a 360° image of the environment. The camera faces upwards with a cone mirror on it. The computer system uses Intel NUC6i7KYK as the main processor, which functions to process data from sensors as self-localization systems and navigation systems. The microcontroller system functions as an interface between the computer system with the gyrocompass and ball sensors. The microcontroller also functions as an interface between the computer and the actuator system, which includes the ball kicker, the ball catcher, and the omni-wheel drivers.

The IRIS Robot was developed for the RoboCup competition and wheeled soccer robot research such as in [22, 23]. Fig. 2 shows the functional diagram of the IRIS robot. There are two main sensors of the robot, namely the omni-vision camera and the gyrocompass sensor. The output of the omni-vision camera is a 360° image called I_o . The output of the gyrocompass sensor is a robot heading angle called θ_{sz} . θ_o is the offset angle required to direct the robot's coordinates to match the direction of the field's coordinates. θ_{fR} is obtained by adding θ_{sz} to θ_o . The self-localization system then calculates matrix S based on I_o and θ_{fR} , where S is defined as follows:

$$S = [x_{fR} \ y_{fR}]^T \tag{1}$$

where x_{fR} and y_{fR} are the coordinates of the robot on the soccer field coordinate system. S and θ_{fR} then become the input of the robot navigation system and

are used to decide the direction and movement of the robot. The robot navigation system calculates the speed of the robot relative to the soccer field, called N , which is defined as

$$N = [\dot{x}_{fR} \ \dot{y}_{fR} \ \dot{\theta}_{fR}]^T \tag{2}$$

where \dot{x}_{fR} is the speed of the IRIS robot in the x_f -axis direction, \dot{y}_{fR} is the speed of the IRIS robot in

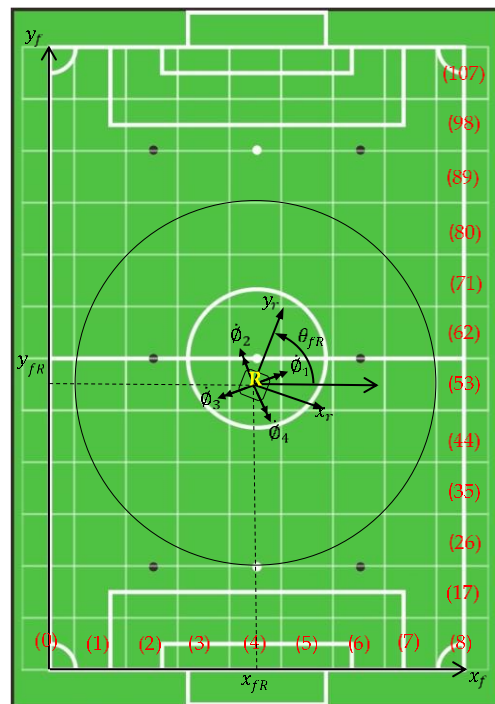


Figure. 3 The soccer field coordinate system (x_f, y_f) and the IRIS robot coordinate system (x_r, y_r) : x_f and y_f are the axes of the soccer field coordinate system; x_r and y_r are the axes of the IRIS robot coordinate system; $R = [x_{fR} \ y_{fR}]^T$ is the position of the robot in the soccer field coordinate system; (0) to (107) are the class numbers of classification of the soccer field

the y_f -axis direction, and $\dot{\theta}_{fR}$ is the angular speed of the IRIS robot. This information is then decoded by the omni-directional wheel drivers to drive the wheels of the robot at the respective speeds of $\dot{\phi}_1, \dot{\phi}_2, \dot{\phi}_3,$ and $\dot{\phi}_4$. A representation of the soccer field coordinate system and robot coordinate system is shown in Fig. 3. The width of the soccer field in parallel with the y_f -axis is 6 m, and the length of the soccer field in parallel with the x_f -axis is 9 m. The soccer field is divided into 108 areas for the classification of robot positions. These classes have a class number from 0 to 107, and every class has an area of 0.5625 m².

2.1 Omni-vision camera

The first sensor of the IRIS robot is an omni-vision camera [24]. The omni-vision camera used is the Logitech 922 camera, which faces upwards to a mirror on top. This omni-vision camera system is very efficient in the application of wheeled soccer robots, as presented by Ant3nio et al. [25]. The convex mirror on top of the camera projects an image of the field that is captured by the camera. An acrylic

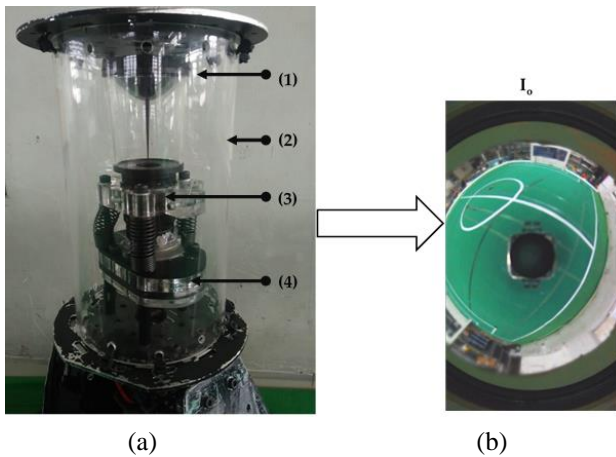


Figure. 4 Omni-vision camera for capturing 360° images: (a) mechanical construction of the omni-vision camera: (1) cone mirror, (2) acrylic tube as protection support, (3) focus lens, (4) the camera, and (b) the 360° image.

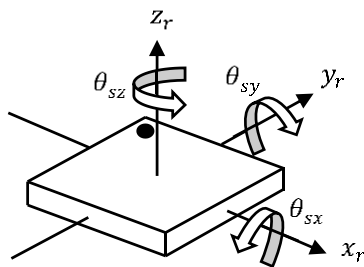


Figure. 5 MPU 6050 gyro compass sensor axis orientation; x_r, y_r, z_r are the axes of the robot coordinate system; $\theta_{sx}, \theta_{sy}, \theta_{sz}$ are the angle of the robot body with respect to its axis

tube is used to strengthen the structure of the camera system that is resistant to impacts of balls or other objects. The camera is placed at the bottom and held by a protective bracket.

The camera system and 360° images are shown in Fig. 4. Io is a 360° image captured by the camera and shown in Fig. 4 (b). This image has a resolution of 360 × 640 pixels, which then becomes the input for pre-processing.

2.2 Gyrocompass sensor

The second sensor of the IRIS robot is the gyrocompass sensor. The robot uses the MPU6050 sensor manufactured by InvenSense Inc. This sensor is a six-axis motion sensor derived from three-axis Micro-Electro Mechanical Systems (MEMS) gyroscope rate sensors with 16-bit Analog-to-Digital Converters (ADCs) and three-axis MEMS accelerometer sensors with 16-bit ADCs [26]. The output of this sensor is sent to the microcontroller using I2C communication. The gyrocompass sensor axis orientation is shown in Fig. 5. Because the robot only moves on a two-dimensional plane in the direction of the x_r -axis and y_r -axes, the θ_{sy} and θ_{sx} angles can be ignored. The θ_{sz} will give a positive value when the robot rotates clockwise.

2.3 Self-localization

This paper uses a self-localization method based on image information from a 360° camera. This method satisfies the robot’s requirement of real-time position information on the field. This method has been widely developed by wheeled soccer robot researchers, such as in [27] using landmarks on the soccer field. Further development was also carried out in [28] by using particle filters. Development was also carried out in [8] by using a deep neural network to detect objects around a mobile robot. This research uses a convolutional neural network method that is specifically used in wheeled soccer robot applications. A discussion of the proposed method will be presented in section 3.

2.4 Robot navigation

The role of the robot navigation system is to regulate the movement of the robot from one point to another. This movement is controlled by the algorithm of the robot. In this paper, the robot navigation system is only used for the movement of robots in data retrieval. The robot navigation system uses inputs of $x_{fR}, y_{fR},$ and θ_{fR} to move the robot to a certain point, then produces outputs of $\dot{x}_{fR}, \dot{y}_{fR},$

and $\dot{\theta}_{fR}$. These outputs are then inputted to the omnidirectional wheel drivers.

2.5 Gyrocompass sensor

The output of the robot navigation system is the speeds \dot{x}_f , \dot{y}_f , and $\dot{\theta}_f$ that are assigned in the field coordinate system, while the movement of the four wheels is assigned in the robot coordinate system. This requires a transformation from the field coordinate system to the robot coordinate system. An explanation of these two coordinate systems is illustrated in Fig. 6. There are two coordinate systems in Fig. 6, namely the robot coordinate system and the soccer field coordinate system. x_r and y_r are the axes of the robot coordinate system, and x_f and y_f are the axes of the soccer field coordinate system. The transformation of the soccer field coordinate system to the robot coordinate system is shown in the following formula:

$$\begin{bmatrix} \dot{x}_{rR} \\ \dot{y}_{rR} \\ \dot{\theta}_{rR} \end{bmatrix} = \begin{bmatrix} \sin(\theta_{fR}) & -\cos(\theta_{fR}) & 0 \\ \cos(\theta_{fR}) & \sin(\theta_{fR}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x}_{fR} \\ \dot{y}_{fR} \\ \dot{\theta}_{fR} \end{bmatrix} \quad (3)$$

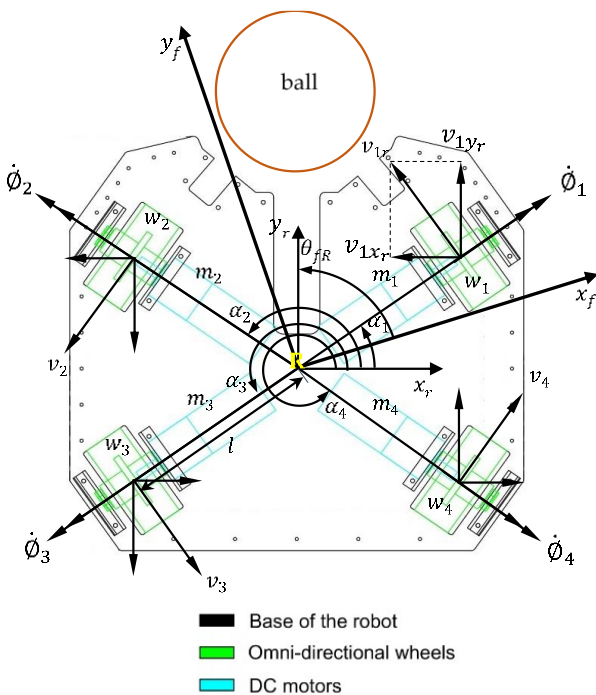


Figure. 6 Four wheels omnidirectional drive actuator: R is the center point of the robot; m_{1-4} are $motor_{1-4}$; w_{1-4} are $wheel_{1-4}$; θ_{fR} is the angle of the robot to the soccer field; α_{1-4} gives the angle between y_r to the axis of m_{1-4} ; v_{1r-4r} gives the linear velocity of w_{1-4} ; and l is the length of the robot center to the center of each wheel

where \dot{x}_{rR} and \dot{y}_{rR} are the linear speed of the robot on the robot coordinate system in m/s, $\dot{\theta}_{fR}$ is the angular speed of the robot in rad/s, θ_{fR} is the angle of the robot to the soccer field in radian, \dot{x}_{fR} and \dot{y}_{fR} are the linear speed of the robot on the soccer field coordinate system in m/s.

There are four motors used to drive the omnidirectional wheels, which function to move and orientate the robot to a determined point in the field. The calculation of the linear speed of the motors can be performed with the following formula:

$$\begin{bmatrix} v_{1r} \\ v_{2r} \\ v_{3r} \\ v_{4r} \end{bmatrix} = \begin{bmatrix} -\sin(\alpha_1) & \cos(\alpha_1) & l \\ -\sin(\alpha_2) & \cos(\alpha_2) & l \\ -\sin(\alpha_3) & \cos(\alpha_3) & l \\ -\sin(\alpha_4) & \cos(\alpha_4) & l \end{bmatrix} \begin{bmatrix} \dot{x}_{rR} \\ \dot{y}_{rR} \\ \dot{\theta}_{rR} \end{bmatrix} \quad (4)$$

where v_{1r-4r} give the tangential speed of motors in m/s, α_{1-4} gives the angle between the x_r -axis and the axis of the motor, \dot{x}_{rR} and \dot{y}_{rR} are the linear speed of the robot in m/s, and $\dot{\theta}_{rR}$ is the angular speed of the robot in rad/s. After obtaining the tangential speed of each motor, the angular speed of each wheel is calculated using the following formula:

$$\begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \\ \dot{\phi}_4 \end{bmatrix} = \begin{bmatrix} v_1/r_1 \\ v_2/r_2 \\ v_3/r_3 \\ v_4/r_4 \end{bmatrix} \quad (5)$$

where $\dot{\phi}_{1-4}$ gives the angular speed of wheels in rad/s and r_{1-4} gives the radius of wheels in meters.

3. Proposed self-localization method

The focus of this paper is the self-localization method using the input of the Omni-vision image and angle of the robot on the field. The proposed process is illustrated in Fig. 7. I_o is an image from the omnivision camera as an input for image pre-processing. Image pre-processing generates a region of interest of the image, which separates the area inside the soccer field and the area outside the soccer field. This region of interest is important because the information outside the soccer field cannot be used as a feature in the self-localization process. The features that are used as references must be features in the soccer field because, in real conditions, the area outside the soccer field will be very different for different venues. The output from pre-processing is the soccer field image, named I_p .

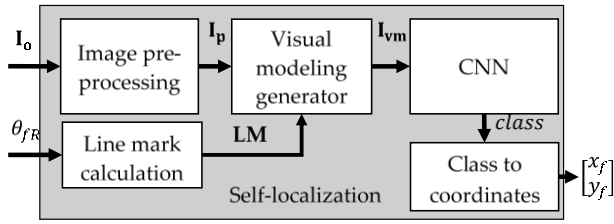


Figure. 7 Block diagram of proposed self-localization method. CNN: convolutional neural network; LM: line mark

The line mark calculation is the algorithm to calculate **LM** from the input θ_f . The **LM** is defined as

$$\mathbf{LM} = \begin{bmatrix} x_c & y_c \\ x_e & y_e \end{bmatrix} \quad (6)$$

where $[x_c \ y_c]$ is the start point of the line mark and $[x_e \ y_e]$ is the endpoint of the line mark. The visual modeling generator generates I_{vm} from the inputs of I_p and **LM**. The I_{vm} then goes to the CNN system and generate the *class*. The *class* represents a position in the soccer field, as seen in Fig. 3. The class is then converted into a soccer field coordinate as $[x_f \ y_f]^T$. Detailed explanations for each process will be discussed in the next sub-section.

3.1 Image pre-processing

The primary purpose of image pre-processing is to obtain the region of interest from a 360° image. The input and output images of pre-processing are shown in Fig. 8. The first step of image pre-processing is to create a working frame region of interest by making a circle in the middle of the image and a circle on the edge of the image. This process aims to get rid of parts that do not have an important information value. The next step is to change the color system of the Blue-Green-Red (BGR) image to Hue-Saturation-Value (HSV). This process aims to obtain images that are more resistant to changes in light. In the HSV format, thresholding is performed to separate the green soccer field area from the outside area. From the thresholding process, we obtain several areas called contours. The contour areas are combined with the convex hull process. The soccer field area is the result of the convex hull with the largest area, which is then used as a masking for the next process.

The initial image, with a resolution of 360×640 pixels, is then cut into a rectangle with a resolution of 360×360 and cut again to obtain an image of the

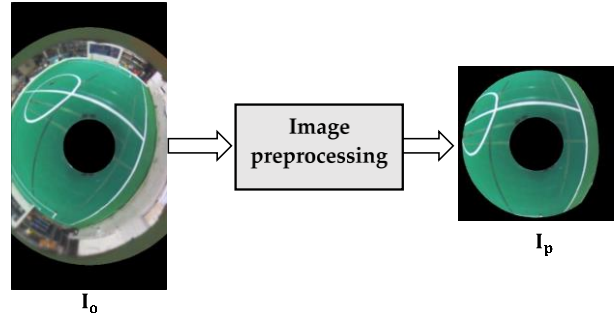


Figure. 8 Image preprocessing input and output images: I_o is the image input; I_p is the image output

center circle with a diameter of 360 pixels. The result of preprocessing is the image named I_p image, as shown in Fig. 8.

3.2 Line mark calculation

Line mark calculation is a process used to calculate the start point and endpoint of the line mark as an input for the visual modeling generator. This process uses the robot angle from the gyrocompass sensor. The first process is performed to determine the start point of the line. The start point of the line mark is the center point of the image, while the endpoint of the line mark is determined based on the robot angle θ_f and the length of the line mark l_{ce} . The robot angle is obtained by the following formula:

$$\theta_f = \theta_o - \theta_{sz} \quad (7)$$

where θ_{sz} is the output of the gyrocompass sensor on the z-axis. This value is independent of the robot's orientation to the soccer field when the sensor is turned on. θ_o is the offset value that is manually obtained by pressing a button when the robot is in line with y_f -axis. The length of l_{vm} is determined beforehand, and this paper uses four different length variations of 40, 80, 120, and 160 pixels. The second

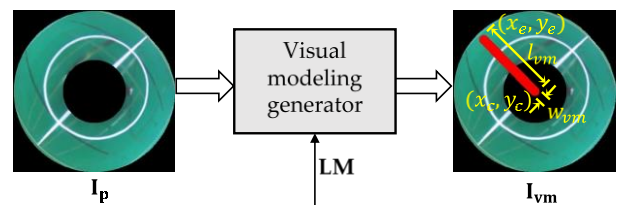


Figure. 9 Visual modelling generator: I_p is the input image; I_{vm} is the output image: (x_c, y_c) is the start point of the line mark as well as the center of the image; (x_e, y_e) is the endpoint of the line mark; l_{vm} is the length of the line mark in pixels; w_{vm} is the width of the line mark in pixels

process is performed to determine the endpoint of the line by the following formula:

$$\begin{bmatrix} x_e \\ y_e \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \end{bmatrix} - l_{vm} \times \begin{bmatrix} \cos(\theta_f) \\ \sin(\theta_f) \end{bmatrix} \quad (8)$$

where $[x_e \ y_e]^T$ is the endpoint of the visual modeling line mark, and $[x_c \ y_c]^T$ is the start point of the line mark, which is also the center of the image.

3.3 Visual modeling generator

The output image I_p from image pre-processing is an RGB image with a resolution of 360×360 pixels. This image is the input image for the visual modeling generator. Another input for the visual modeling generator is LM , which consists of the start point and the endpoint of the line mark. This line mark is generated by the following algorithm:

$$I_{vm} = \text{line}(I_p, \text{Point}(x_c, y_c), \text{Point}(x_e, y_e), \text{Scalar}(0, 0, 255), w_{vm}) \quad (9)$$

where line is the OpenCV library [29] to draw a line with a start coordinate of $\text{Point}(x_c, y_c)$, and end coordinate of $\text{Point}(x_e, y_e)$; $\text{Scalar}(0, 0, 255)$ is the BGR color parameter with (B = 0, G = 0, R = 255); and w_{vm} is the width of the line mark in pixels. The input and output images of the visual modeling generator are illustrated in Fig. 9. The result of this process is I_{vm} , which then becomes an input to the CNN process. This paper changes the values of l_{vm} and w_{vm} to get the best CNN model for self-localization applications in wheeled soccer robots.

3.4 Convolutional neural network

The proposed CNN architecture used in this paper is shown in Fig. 10. This architecture was chosen based on several trials that had been performed before. The first convolutional filter layer has an input image with a resolution of 360×360 . $W_{in} = 360$, $H_{in} = 360$, and $C_{in} = 3$, where W_{in} is the width of the input image, H_{in} is the height of the input image, and C_{in} is the color dimension of the input image. The first convolutional filter layer has the parameters $K = 32$, $F = 4$, $S = 1$, and $P = 0$, where K is the number of filters, F is the filter dimension, S is the stride, and P is the padding. The amount of output from this filter is calculated with the following formula:

$$W_{out} = \frac{W_{in} - F + 2P}{S} + 1 \quad (10)$$

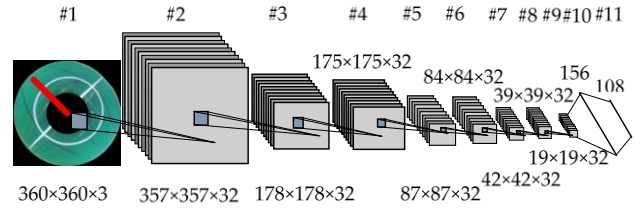


Figure. 10 Proposed convolutional neural network for wheeled soccer robot application

$$H_{out} = \frac{H_{in} - F + 2P}{S} + 1 \quad (11)$$

$$C_{out} = K \quad (12)$$

where W_{out} , H_{out} , and C_{out} are the output dimensions of the first convolution process. This parameter applies to all convolution processes from conv2d_1 to conv2d_4. For max-pooling, we use a 2×2 size, with stride = 2. This process will produce an output size of half the size of the input. We use four max-pooling processes from max_pooling2d_1 to max_pooling2d_4.

The last part of CNN for localization is the fully connected layer, which consists of two layers, each with 256 nodes and 108 nodes. This last node is the output of 108 classes that is used to divide areas in the soccer field. The activation function of this last node is the softmax function, which produces a value of 1 if all the outputs of the nodes are added. The node with the largest value indicates the output class value from the CNN system. The implementation of CNN was built with the Keras library with a Tensorflow-GPU back-end [30]. The implementations of the CNN architecture are coded using Python.

3.5 Class to soccer field coordinate conversion

The CNN system produces an output in the form of classes from input images. This class is a representation of the position of the robot in the soccer field in accordance with the division of the field cluster in Fig. 3. The value in this class is then transformed to the value of the soccer field coordinates with the following formula:

$$x_{fR} = (\text{class} \bmod 9) \times 75 \quad (13)$$

$$y_{fR} = (\text{floor}(\text{class} / 9) \times 75) + 37.5 \quad (14)$$

where mod is the modulo operation, floor is the round down operation, and x_{fR} and y_{fR} are robot coordinate points as the outputs of self-localization.

4. Experiment and results

This chapter discusses the results of the data sets that have been made and the results of CNN tests that have been carried out. In this paper, we produce a data set that can be used for wheeled soccer robot applications. This data set can also be used for other purposes, such as processing visual data related to the localization of wheeled soccer robots on a playing field. The results of the CNN test are CNN architecture recommendations and a comparison of the proposed method with conventional CNN methods. Several comparisons of the length and width of visual modeling lines have also been made, and recommendations for the best results are given.

4.1 Experimental setup for dataset retrieval

This paper presents a new data set in the form of omni-vision images on a soccer field taken by the IRIS robot. The images are grouped into 108 classes that are tagged with robot positions taken from a global camera.

The data set setup is depicted in Fig. 11. The global camera is placed above the soccer field at a distance of 4 m. This camera uses a wide lens that is able to capture the entire area of the soccer field. The playing field is made of green carpet with white soccer field lines. In this paper, we use a soccer field size of 9 m × 12 m in accordance with the standards of the Indonesian national competition of wheeled soccer robots.

The block diagram of the data set recorder is shown in Fig. 12. The global camera captures images of the soccer field, I_g . The I_g image is then processed on the base station computer using the Neural Network method to get the robot's coordinates on the soccer field, called x_{fR} and y_{fR} . These coordinates are sent to the IRIS robots using a WiFi network. The IRIS robot moves on the soccer field following a

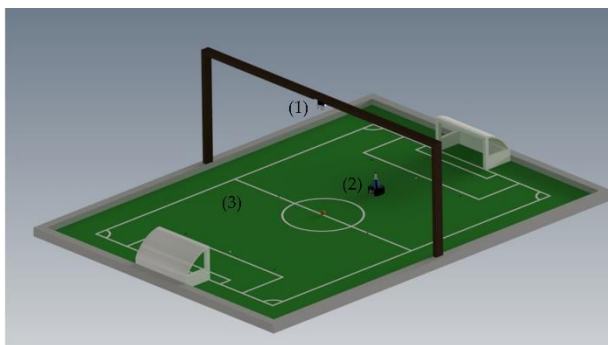


Figure. 11 Data set recorder setup: (1) the global camera for capturing the position of the robot on the soccer field, (2) the IRIS Robot with an on-board omni-vision camera, and (3) the soccer field with a green-colored floor and white-colored lines

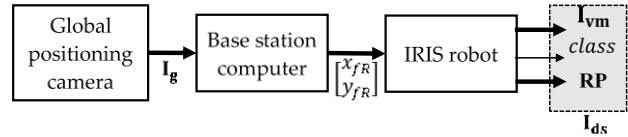


Figure. 12 Block diagram of the data set recorder: I_g is the image from the global positioning camera; I_{ds} is the omni-vision data set image produced by the IRIS robot, consisting of I_{vm} as a 360° image with gyrocompass line mark visual modeling, $class$ as the cluster of the robot position on the soccer field, and $RP = [x_{fR} \ y_{fR} \ \theta_{fR}]^T$ as the robot position coordinate on the soccer field

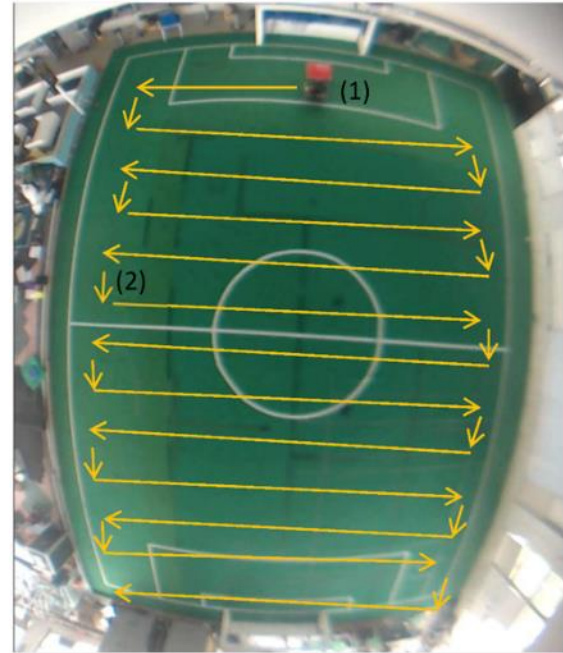


Figure. 13 Retrieval path of data set recording: (1) IRIS robot and (2) path plan for data retrieval

certain route while recording omni-vision images and the coordinates of x_{fR} , y_{fR} , and θ_{fR} . The global coordinate index is written for each frame of the recorded image. The assigned route for robot movement seen from a global camera is drawn as in Fig. 13.

4.2 Data classification and labeling

The image recording results are stored in a video file with global coordinate data assigned for each frame. Global coordinate data are stored in a text file using the CSV format. Images are classified into 108 classes according to their tagged position. The classification formula is seen in the following formula:

$$class = floor\left(\left(\frac{y_{fR}-37.5}{75} \times 9\right) + \frac{x_{fR}}{75}\right) \quad (15)$$

where *class* is the cluster of classification, *floor()* is a C function to round down the value of float number, x_{fR} and y_{fR} are the coordinates of the robot position on the soccer field, constant number 37.5 is the center of classification box with a length of 37.5 cm, constant number 75 is the length of each classification box with a length of 75 cm, and constant number 9 is the number of columns for each row of the classification box on the soccer field.

Each class is separated in a folder with code labels ranging from 000 to 107. Two data sets are provided that can be used for training and testing. There are 100,008 images belonging to 108 classes for training and 10,419 images belonging to 108 classes for testing.

4.3 CNN training and implementation

The experiment was done by conducting training on the CNN system with the training dataset. The first training was performed without the visual modeling method. The data set of 100,008 images were divided into 90% for training and 10% for validation. The training process used an Intel i7 processor computer with a Nvidia Geforce GTX 1070 graphics card. For the training process, we observed the loss values to stop the training process. If the change in loss was smaller than 0.001 for 5 epochs, the training was stopped, and the CNN model with the smallest loss was then used.

In the first training without visual modeling lines, the results were $acc = 0.9908$, $loss = 0.0303$, $val_acc = 0.7007$, and $val_loss = 1.4478$, where *acc* is the accuracy of training data for each epoch, *loss* is the loss of training data set for each epoch, *val_acc* is the accuracy of the validation dataset for each epoch, and *val_loss* is the loss of validation dataset for each epoch. This training stopped at 36 epochs with a training time of 332 min, but the smallest loss was obtained at 33 epochs. The second training used the CNN method in accordance with the proposed method. In this training, variations in the length and width of the visual modeling line were used. The first length and width used were 5 pixels wide and 40 pixels long. The results of this training were $acc = 0.9903$, $loss = 0.0310$, $val_acc = 0.6620$, and $val_loss = 1.6772$. Training as further conducted by changing the width and length values of the visual modeling lines. The results of the training can be seen in Table 1. The average learning process obtained the smallest loss at 38 epochs with an accuracy of more than 0.99 and an average loss of 0.0219. The average accuracy with validation data was 0.7768, and the average loss of validation data was 1.0131.

Table 1. Training result of proposed data set for training with variations of the width and length of lines marking visual modeling

No	Line Mark Width × Length (pixels)	Epoch	acc*	loss	val_acc**	val_loss***
1	No line	33	0.990	0.030	0.700	1.447
2	5×40	48	0.994	0.019	0.793	1.044
3	5×80	33	0.993	0.020	0.757	0.961
4	5×120	35	0.992	0.024	0.783	0.878
5	5×160	22	0.993	0.021	0.797	1.061
6	10×40	42	0.995	0.014	0.802	0.961
7	10×80	33	0.992	0.022	0.791	0.970
8	10×120	60	0.993	0.029	0.784	1.079
9	10×160	30	0.992	0.023	0.768	0.944
10	15×40	42	0.993	0.022	0.789	0.831
11	15×80	35	0.993	0.022	0.751	0.897
12	15×120	28	0.992	0.025	0.781	0.809
13	15×160	40	0.994	0.020	0.797	1.068
14	20×40	39	0.994	0.019	0.803	0.909
15	20×80	49	0.994	0.017	0.800	1.009
16	20×120	41	0.993	0.019	0.741	1.287
17	20×160	38	0.991	0.026	0.764	1.059

*accuracy; **validation of accuracy; ***validation of loss

The results of the CNN model were then implemented by the IRIS robot system using the testing dataset. Testing was done for each model that was obtained in the training process. Without using the gyrocompass visual modeling marking line, the CNN system had difficulty recognizing the position of the robot on the soccer field and only obtained an accuracy of 0.3262. This is because the lines on the soccer field are a reflection of the x_f -axis and y_f -axis. By using visual modeling, the results obtained were significantly improved. The accuracy value comparison obtained from the CNN system with variations of the width and length of the visual modeling marking line can be seen in Table 2. Generic or Conventional CNN is employed without line mark as noted as $l_{vm} = 0$ px in Table 2, same method used in [8][9] but different application, which is for indoor house localization, not for soccer field localization with very few features.

A maximum accuracy of 0.6827 was obtained at a width of 20 pixels and a length of 160 pixels. The addition of width in the visual modeling line mark tends to increase the accuracy of the CNN system, but

Table 2. Accuracy comparison of variations in w_{vm} and l_{vm} of the gyrocompass line mark visual modeling. Conventional CNN (0 px) method for $l_{vm}=0$ and proposed method for for $l_{vm}>0$

w_{vm} \ l_{vm}	Conv. CNN	Proposed Method				
	0 px	40 px	80 px	120 px	160 px	
5px	0.3262	0.6525	0.6514	0.6688	0.6687	
10px	0.3262	0.6710	0.6818	0.6520	0.6621	
15px	0.3262	0.6795	0.6245	0.6349	0.6803	
20px	0.3262	0.6725	0.6636	0.6379	0.6827	

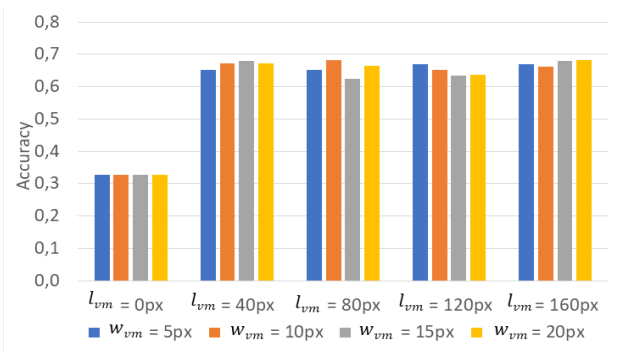


Figure. 14 Accuracy comparison chart with variations in width and length of gyrocompass line mark visual modeling. Conventional CNN method for $l_{vm}=0$ and proposed method for for $l_{vm}>0$

this does not apply to all conditions. For example, at a length of 120 pixels, the addition of width decreases accuracy. The addition of length in the visual modeling line tends to increase the accuracy of the CNN system. An accuracy comparison chart with variations in the width and length of the gyrocompass line mark visual modeling can be seen in Fig. 14. In addition, the CNN is an end-to-end implementation used directly to do self-localization that has an output of the robot position. It is different and more robust compare with work [11] that use CNN for pre-image pre-process acceleration for robot detection and localization.

5. Conclusions

A CNN-based self-localization method using the visual modeling of the gyrocompass line mark and omni-vision image for wheeled soccer robots is proposed. This new method was used to improve the accuracy of conventional CNN-based self-localization for wheeled soccer robot applications. This proposed method is able to increase the conventional CNN accuracy from 0.3262 to 0.6827. The proposed system improves the accuracy by 0.3565 from the comparison method.

The dataset retrieval system using a global camera can accelerate the process of tagging the soccer field coordinates of the robot. The location of the robot on the soccer field and the classification of the robot’s area on the field can be done quickly and automatically. The new original dataset generated in this paper is expected to be used for other purposes related to the development of localization techniques in wheeled soccer robots.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization, R.D.; methodology, R.D.; software, R.D.; validation, R.D., and S.; formal analysis, R.D.; investigation, R.D., and S.; resources, R.D.; data curation, R.D., and S.; writing—original draft preparation, R.D.; writing—review and editing, D.P., and T.S.; visualization, R.D.; supervision, D.P., and T.S.;

Acknowledgments

Acknowledgment is given to LPDP (Indonesia Endowment Fund for Education) with the BUDI DN scheme [PRJ-5461 /LPDP.3/2016], which has provided a scholarship for the author’s continued education at the Dept. of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia.

References

- [1] H. Kitano, M. Asada, I. Noda, and H. Matsubara, “RoboCup: robot world cup”, *IEEE Robot. Autom. Mag.* Vol. 5, pp. 30–36, 1998.
- [2] Setiawardhana, R. Dikairono, D. Purwanto, and T. A. Sardjono, “Ball Position Estimation in Goal Keeper Robots Using Neural Network”, *International Review of Automatic Control (I.R.E.A.CO.)*, Vol. 13, No. 1, 2020.
- [3] R. H. Abiyev, N. Akkaya, and I. Günsel, “Control of Omnidirectional Robot Using Z-Number-Based Fuzzy System”, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 49, No. 1, 2019.
- [4] MD. A. Al Mamun, M.T. Nasir, and A. Khayyat, “Embedded System for Motion Control of an Omnidirectional Mobile Robot”, *IEEE Access*, Vol. 6, pp. 6722-6739, 2018.
- [5] S. A. Magalhães, A. P. Moreira, and P. Costa, “Omnidirectional robot modeling and simulation”, In: *Proc. of 2020 IEEE*

- International Conf. on Autonomous Robot Systems and Competitions (ICARSC)*, Ponta Delgada, Portugal, pp. 251-256, 2020.
- [6] M. Fuad, T. Agustinah, and D. Purwanto, "Collision Avoidance of Multi Modal Moving Objects for Mobile Robot Using Hybrid Velocity Obstacles", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No.3, pp.407-421, 2020.
- [7] R. Watiasih, M. Rivai, O. Penangsang, F. Budiman, Tukadi, and Y. Izza, "Online Gas Mapping in Outdoor Environment using Solar-Powered Mobile Robot", In: *Proc. of 2018 International Conf. on Computer Engineering, Network and Intelligent Multimedia (CENIM)*, Surabaya, Indonesia, pp. 245-250, 2018.
- [8] S. Nilwong, D. Hossain, S. Kaneko, and G. Capi, "Deep Learning-Based Landmark Detection for Mobile Robot Outdoor Localization", *Machines*, Vol. 7, No.25, 2019.
- [9] S. P. P. da Silva, R. V. M. da Nóbrega, A. G. Medeiros, L. B. Marinho, J. S. Almeida, and P. P. R. Filho, "Localization of Mobile Robots with Topological Maps and Classification with Reject Option using Convolutional Neural Networks in Omnidirectional Images", In: *Proc. of International Joint Conf. on Neural Networks (IJCNN)*, Rio de Janeiro, pp. 1-8, 2018.
- [10] L. B. Marinho, J. S. Almeida, J. W. M. Souza, V. H. C. Albuquerque, and P.P.R. Filho, "A novel mobile robot localization approach based on topological maps using classification with reject option in omnidirectional images", *Expert Systems with Applications*, Vol. 72, pp. 1-17, 2017.
- [11] S. Luo, H. Lu, J. Xiao, Q. Yu, and Z. Zheng, "Robot detection and localization based on deep learning", In: *Proc. of 2017 Chinese Automation Congress (CAC)*, Jinan, pp. 7091-7095, 2017, doi: 10.1109/CAC.2017.8244056.
- [12] N. Chang, Y. Chien, H. Chiang, W. Wang, and C. Hsu, "A Robot Obstacle Avoidance Method Using Merged CNN Framework", In: *Proc. of 2019 International Conf. on Machine Learning and Cybernetics (ICMLC)*, Kobe, Japan, pp. 1-5, 2019.
- [13] S. Hsu, Y. Wang, and C. Huang, "Human Object Identification for Human-Robot Interaction by Using Fast R-CNN", In: *Proc. of 2018 Second IEEE International Conf. on Robotic Computing (IRC)*, Laguna Hills, CA, pp. 201-204, 2018.
- [14] R. Jiménez-Moreno and J. O. Pinzón-Arenas, "New Hybrid Fuzzy-CNN Architecture for Human-Robot Interaction", *International Review of Automatic Control (IREACO)*, Vol. 12, No. 5, pp. 236–241, 2019.
- [15] T. Mehta and C. Bhensdadia, "Adaptive Near Duplicate Image Retrieval Using SURF and CNN Features", *International Journal of Intelligent Engineering and Systems*, Vol. 12, No. 5, pp. 104-115, 2019.
- [16] V. Campos, B. Jou, and X. Giró-i-Nieto, "From pixels to sentiment: Fine-tuning CNNs for visual sentiment prediction", *Image and Vision Computing*, Vol. 65, No. C, 2017.
- [17] T. Chen, R. Xu, Y. He, and X. Wang, "Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN", *Expert Systems with Applications*, Vol. 72, pp. 221–230, 2017.
- [18] M. Elleuch, R. Maalej, and M. Kherallah, "A New Design Based-SVM of the CNN Classifier Architecture with Dropout for Offline Arabic Handwritten Recognition", *Procedia Computer Science*, Vol. 80, pp. 1712–1723, 2016.
- [19] K. Kamnitsas, C. Ledig, V. F. J. Newcombe, J. P. Simpson, A. D. Kane, D. K. Menon, D. Rueckert, and B. Glocker, "Efficient multi-scale 3D CNN with fully connected CRF for accurate brain lesion segmentation", *Medical Image Analysis*, Vol. 36, pp. 61–78, 2017.
- [20] K. Watanabe, Y. Ma, and H. Suzuki, "Real-time Self-localization using Model-based Matching for Autonomous Robot of RoboCup MSL," *Journal of Robotics, Networking and Artificial Life*, Vol. 7, No.2, pp. 98-102, 2020.
- [21] E. A. Nurrohmah, B. S. Bayu, M. M. Bachtiar, I. K. Wibowo, and R. Adryantoro, "Detecting Features of Middle Size Soccer Field using Omnidirectional Camera for Robot Soccer ERSOW", In: *Proc. of International Conf. on Smart Technology and Applications (ICoSTA)*, Surabaya, Indonesia, pp. 1-6, 2020.
- [22] M. Asada, H. Kitano, I. Noda, and M. Veloso, "RoboCup: Today and tomorrow – What we have learned", *Artif. Intel.*, 110, pp.193-214, 1999.
- [23] M. J. G. van de Molengraft and O. Zeigle, "Special issue on Advances in intelligent robot design for the RoboCup Middle Size League", *Mechatronics*, Vol. 21, pp. 399-410, 2011.
- [24] C. Jaramillo, R. G. Valenti, L. Guo, and J. Xiao, "Design and Analysis of a Single—Camera Omnistereo Sensor for Quadrotor Micro Aerial Vehicles (MAVs)", *Sensors*, Vol. 16, No. 2, pp. 217, 2016.
- [25] A. J. R. Neves, A. J. Pinho, D. A. Martins, and B. Cunha, "An efficient omnidirectional vision system for soccer robots: From calibration to

- object detection”, *Mechatronics*, Vol. 21, pp. 399-410, 2011.
- [26] InvenSense Inc, MPU-6000 and MPU-6050 Product Pedification Revision 3. 4, InvenSense Inc. San Jose, CA, 19 August 2013.
- [27] C. F. Marques and P. U. Lima, “Vision-based self-localization for soccer robots”, In: *Proc. of the 2000 IEEE/RSJ International Conf. on Intelligent Robots and Systems (IROS 2000)*, Takamatsu, Japan, pp. 1193-1198, 2000.
- [28] C. Y. Chen, C. Y. Ho, C. C. Wong, and H. Aoyama, “Omnidirectional vision-based robot localization on soccer field by particle filter”, In: *Proc. of SICE Annual Conf. 2010*, pp. 2976-2981, 2010.
- [29] G. Bradski, “The OpenCV Library”, Dr Dobbs J. Softw. Tools ver 3. 4, 2000.
- [30] F. Chollet, Keras, San Fransisco, U. S., Github, 2015.