Implementation of a Smart System Digital Image Processing of Stair Detection for Covid-19 Robot with Haar-Like Feature Method

Emilliano, D. Sopian, Irma R. Defi, Dessy Novita, Erwin Sitompul, Arjon Turnip*

***Abstract*— During the current Covid-19 pandemic, of course, all activities are limited, thus encouraging people to be more creative and innovative in all fields, especially in the field of technology. Everyone will be encouraged to create the right technology to deal with the pandemic situation in the future. To help break the chain of the spread of the corona virus, the Covid Robot was created. This robot will help the needs of patients who are sick due to covid in the hospital or can help the medical team not to come into direct contact with patients affected by COVID-19. In carrying out its work, of course, the robot needs to recognize several objects that will be encountered in the hospital, so it needs a program to distinguish objects and detect the presence of objects that will produce output recognition of objects in the hospital. With the increasingly sophisticated technology and the stronger computing capabilities of computers, many computational techniques such as artificial intelligence (AI) can be implemented in the real world. By utilizing one technology to solve the problem of detecting down stairs in a hospital, we need image processing technology, this technology is called digital image processing or Digital Image Processing which is part of Machine Learning and also Deep Learning. In digital image processing, we also need to make a command with the Python programming language so that it can be understood by the computer when it will run the given command for down stair Detection using Haar-like Feature.**

***Index Terms*—Image Processing, Stair Detection, Covid-19 Robot, Artificial Intelligence, Haar-Like Feature, Pattern Recognition**

I. Introduction

ALONG with the times and the emergence of the transmission of the Covid-19 Virus called the Covid-19 pandemic, the development of Robots with electric wheels appeared by utilizing LIDAR signals to run automatically for Doctor Aids in carrying medical equipment to carry out health control or consultation with Covid-19 patients or collect patient health data such as heart rate, blood pressure, body temperature, and others without direct exposure to Covid-19 patients in hospitals. Automation-controlled Covid-19 Robot or called Medical Robot - Controlled Intelligent Assistive Technology (MR-CIAT) is an unmanned Autonomous Ground Vehicle (UGV) system integration by utilizing digital cameras and image processing that runs automatically from one patient to another to daily health data collection of Covid-19 patients without the intervention of doctors or nurses (other medical personnel). The LIDAR signal is processed by the Mini PC from the Covid-19 Robot so that it can translate the LIDAR signal into mapping information for the Robot's position to move the electric wheel from Covid-19.

The robot moves from the position of 1 Covid-19 patient to the position of another patient. Over time, concerns have arisen over the safety and comfort of Covid-19 Robot users. This concern arises because MR-CIAT users have difficulty knowing if they are facing stairs going down. Therefore, research was made in this paper that developed the ability of MR-CIAT in order to detect and recognize stairs going down using image processing and pattern recognition with a digital camera. This study uses artificial intelligence based the Haar-Like Feature method with the Python programming language.

Therefore, the design of the MR-CIAT system must be equipped with the ability to detect down stairs obstacles in order to prevent accidents when the MR-CIAT cannot operate properly. Digital image processing is needed to detect obstacles in the form of stairs going down. Image processing using the Open Source Computer Vision Library (OpenCV) in 2013 [1] is carried out to analyze and recognize obstacle objects in real-time. The related topic also has been conducted in our previous research [2-6].

In this study, an obstacle detection system was implemented using the Haar-Like Feature method introduced by Paul Viola and Michael Jones in 2013 [7]. The use of Haar-Like Feature method is generally used for the face detection process, but in this research, the Haar-Like Feature method will be applied to the object detection process in the form of down stairs obstacles. With this method, it is hoped that MR-CIAT can recognize obstacles in the form of stairs going down so that robot can avoid obstacles.

This Research was funded by Indonesia (PPKI) – Indonesian Collaborative Research Program Grant

Emilliano, Dede Sopian, Dessy Novita, Arjon Turnip, Electrical Engineering Department, Universitas Padjadjaran, Jatinangor – Sumedang, Indonesia (E-mails: emilliano@unpad.ac.id, turnip@unpad.ac.id*)

Irma Ruslina Defi, Hasan Sadikin General Hospital, Faculty of Medicine, Universitas Padjadjaran, Indonesia (irma.ruslina@unpad.ac.id).

Erwin Sitompul, Study Program of Electrical Engineering, President University, Cikarang, Indonesia (sitompul@president.ac.id).

II. CONSTRUCTION AND IMAGE PROCESSING

A. Open CV

OpenCV is an open-source program that focuses on real-time computer vision. OpenCV supports C, C++, Python, and Java programming languages that can be applied to operating systems (OS) such as Windows, Linux, Mac OS, iOS, and android [8]. The OpenCV library contains mathematical functions, image processing functions, and functions that include computer vision [9]. In this study, several functions in OpenCV will be used, several functions in OpenCV will be used, including bilateral filter, thresholding, canny edge detection, and haar-like feature.

B. Bilateral Filter

Bilateral Filter is one of the filtering techniques for the image preprocessing stage which was first proposed by C. Tomasi and R. Manduchi in 1998. This filter is a non-linear, edge-preserving, and Gaussian noise-reducing filter that is used for grayscale images or images. colored. The image will be refined but the filter still maintains the sharpness of the image edges. In general, a color image is a combination of three colors: red, green, and blue. If the image color is filtered separately, the smoothness and edges of the image will be different. The smoothness process separately will disturb the color balance and will lead to unexpected color combinations. Therefore, the bilateral filter processes all three colors simultaneously and determines the middle value based on the color match.

The basic concept of a bilateral filter is to create a non-linear combination of similar pixel values. Image filtering uses the calculated value of the range and domain filter. The calculation of the value of the domain filter is determined based on the desired pixel combination. The calculation of the range value is based on the similarity of two pixel values that are determined based on the number of low pass filter values. The low pass filter domain of an image I(x) [10] can be defined as

$$h(x) = kd^{-1}(x) \iint_{-\infty}^{\infty} I(y) * c(x,y)dy \tag{1}$$

with c(x,y) measuring the proximity of the geometric value of the midpoint of x to the neighboring value of y, h(x) is the output image, and (x) is the value of the normalization constant in 2015 [10] which can be calculated as

$$k_d(x) = \iint_{-\infty}^{\infty} c(x,y)dy, \tag{2}$$

and the proximity of the filter range [5] is defined in

$$h(x) = {k_r}^{-1} \iint_{-\infty}^{\infty} I(y) * s(I(x), I(y))dy \tag{3}$$

The measurement of the similarity s(I(x),I(y)) of the photometry value is a light measurement value based on the perception of the brightness level of the human eye between the pixels at the midpoint of x and the point around y. The normalization value in [5] which can be calculated as

$$k_r(x) = \iint_{-\infty}^{\infty} s(I(x), I(y))dy \tag{4}$$

C. Thresholding

Thresholding is the simplest segmentation method. Image segmentation is an important stage in the preprocessing stage. In general, this technique separates the image into 2 parts based on the color intensity. This color separation process is to determine the detection object and the background. The pixel value in an image will be compared with the threshold value, known as the threshold limit in [11]. After the pixels in the image are compared, the object pixel value will be maxVal and the background pixel value will be as shown in Fig. 1.

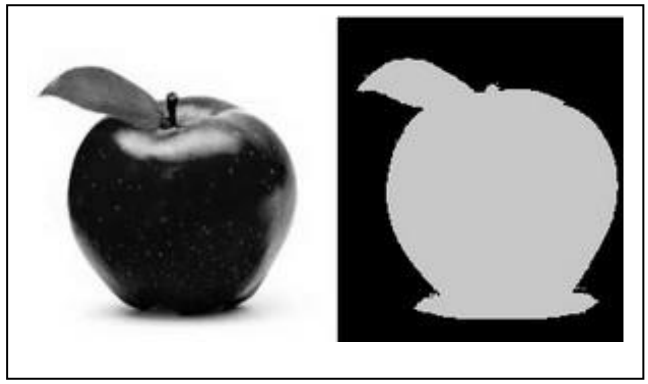


Fig. 1. Original Image and Thresholding Result

The segmentation process using the thresholding method has five types: Thresholding Binary, Threshold Binary (inverted), Truncate, Threshold to Zero, and Threshold to Zero (inverted). Fig. 2 illustrates how the thresholding work process with a blue horizontal line illustrates the threshold value [12].

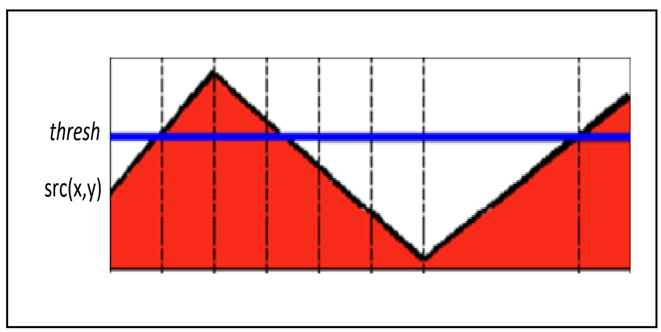


Fig. 2. Thresholding Basic Process [12]

D. Conny Edge Detection

Canny Edge Detection is an edge detection method developed by John F. Canny. Canny edge detection is an algorithm for finding the maximum image gradient value [13]. The process of calculating the image gradient using the Sobel operation. This method uses a 2-stage threshold to detect strong and weak edges and displays weak edges if these edges have a relationship with strong edges [14]. The stages of the canny edge process are shown in Fig. 3.

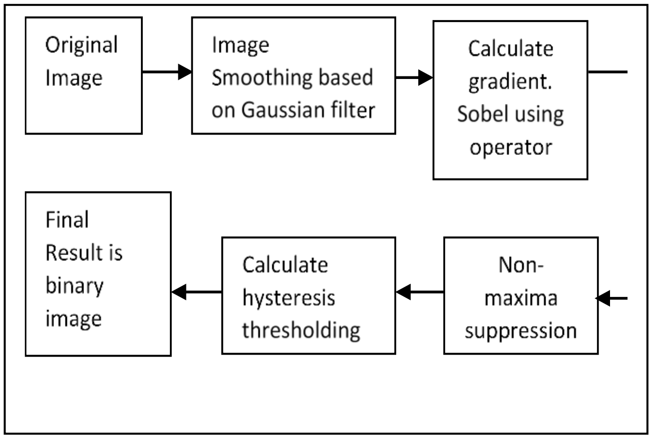


Fig. 3 Canny Edge Process Block [14]

Canny Edge Process Results on ImageThe canny edge algorithm eliminates pixels that are not edges based on the predetermined maxVal and minVal thresholding values. Edge point values that are constrained using a threshold will discard edges that are not related to each other. The results of the canny edge process on the image in the form of stairs in the 30 LIPI building can be seen in Fig. 4.

Fig. 4 Original Image (Left) and Canny Edge Results (Right)

The use of canny edge in OpenCV is as follows:
Python : cv2.Canny(detected_edge, minVal, maxVal)

Parameter:
• detected_edge is the image source
• minVal is the minimum thresholding value
• maxVal is the maximum thresholding value

E. Conny Edge Detection

Haar-like feature algorithm is feature extraction and classification method which was first introduced by Paul Viola and Michael Jones. This method is one of the various effective image recognition methods for detecting 2-dimensional image objects. Haar-like feature information collection is based on light and dark areas in a square area or commonly known as haar-like wavelets [15]. Furthermore, the combination of light and dark pixels will be divided into three features of the square feature of the visual object as shown in Fig. 5.

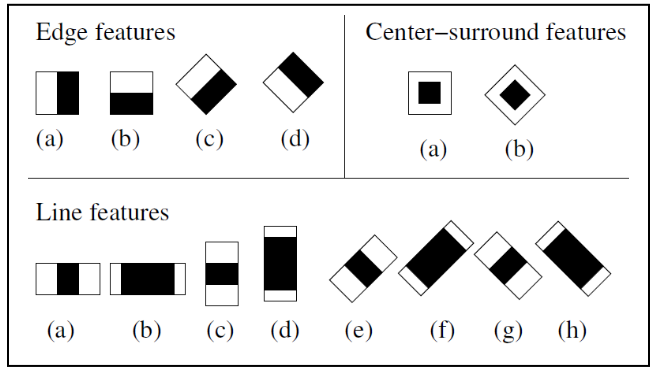


Fig. 5 Haar Feature Variations [15]

Haar features which are divided into dark and light areas will be determined by subtracting the average pixel value in the dark area from the average pixel in the bright area. If the difference value is above the threshold value, then there is a Haar Feature value in the square area. The value of the Haar Like feature is the difference between the number of grayscale pixel values in the black and white squares that have been calculated using the "integral image". The gray level equation for Haar-Like features [15] can be written as in Equation 5.

F. Conny Edge Detection

Adaptive Boosting is a machine learning algorithm formulated by Yoav Freund and Robert Schapire. This algorithm is an adaptive algorithm where the classification of Boosting information is based on the classification of previous information. The cascade classifier results in improved detection performance by decreasing computation time. Boosting classifiers can be created by removing negative images when detecting positive images. Classification is made in stages to get rid of negative images that are not needed in the detection process before detecting more complex objects to get a positive image detection error rate. Fig. 6 describes the levels of the cascade algorithm built by AdaBoost.

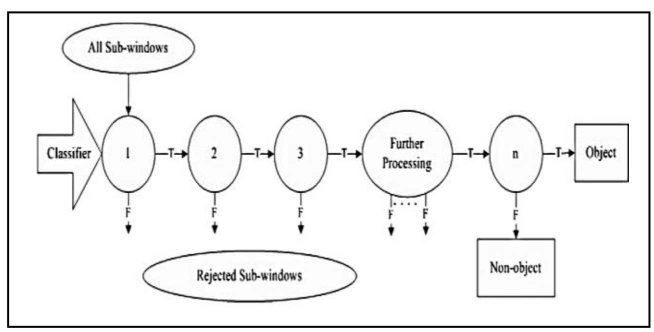


Fig. 6. Cascade Classifier Structure by Ada Boost

III. Research Methods

A. Design and Realization of Digital Image Processing

Digital image processing in this study aims to detect obstacles, namely down stairs. The image obtained is a frame of the surrounding state in real-time. Images with RGB color format will be converted to grayscale first so that image calculations can be accelerated. Image processing will use the openCV library which has image processing functions. The libraries that will be used in this study are bilateral filters to remove noise in the image, canny edge detection to detect image edges, and haar-like features to detect obstacles in the form of descending stairs. Fig. 7 shows the stages of object detection recorded by the camera. Fig. 8 shows the Robot Covid-19 UNPAD.

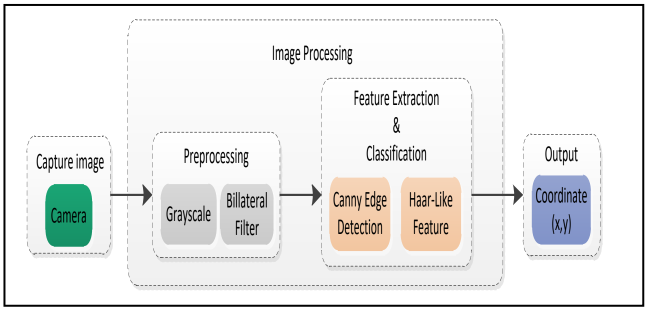


Fig. 7 Image Processing Stages

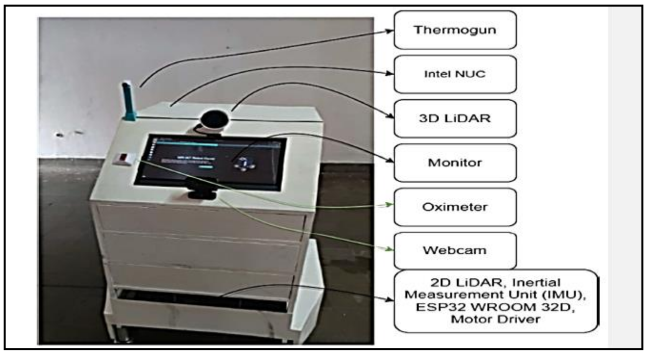


Fig. 8. Robot Covid-19 UNPAD

The initial stage begins with the acquisition of camera frames by Raspicam. Resize is done to reduce the image size to 320x240 pixels. This is done so that the calculation process on an image can be minimized. Fig. 9 is an example of an image input in the form of a ladder recorded using a raspicam.

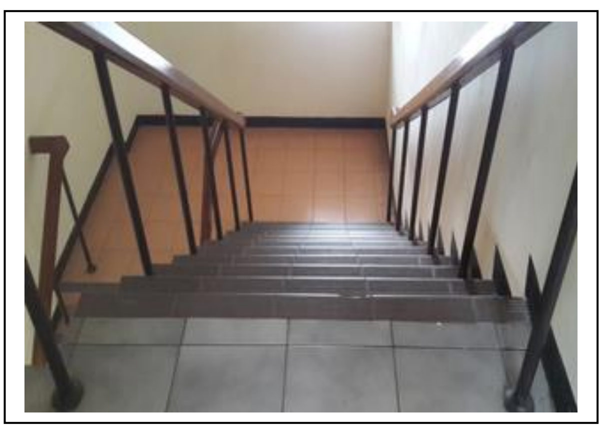


Fig. 9 Input Image

The color format of the image will be changed from RGB to grayscale. Changes in color format are intended to simplify image calculations. If using the RGB format, an image consists of three color components, namely red, green, and blue. So that the image calculation process is carried out 3 times, namely the calculation of the red, green, and blue values. After the three values are calculated, the average value of each pixel will be searched. Meanwhile, when using the grayscale color format, an image will be converted into black and white gradations with a value of 0 to 255. So the calculation process only takes place once. Fig. 10 is the result of the process of converting an RGB image to a grayscale.

Fig. 10. Grayscale Image

The grayscale image format will enter the noise reduction stage using a bilateral filter. The use of a bilateral filter aims to reduce noise in the image by keeping the edges of the image sharp. Because the edges of the image will be used to determine the characteristics of the image in the next canny edge detector method. After the image is filtered, the preprocessing stage of image processing has been completed.

Furthermore, the image will enter the feature extraction stage. At this stage, the characteristics of an image in the form of an obstacle object will be taken and processed with the help of the openCV library. The first method that will be used in this feature extraction process is canny edge detection. This method is used to eliminate unwanted pixel values in the image. The results of the canny edge detection method are the edges of the detected image. The output image of a canny edge can be seen in Fig. 11.

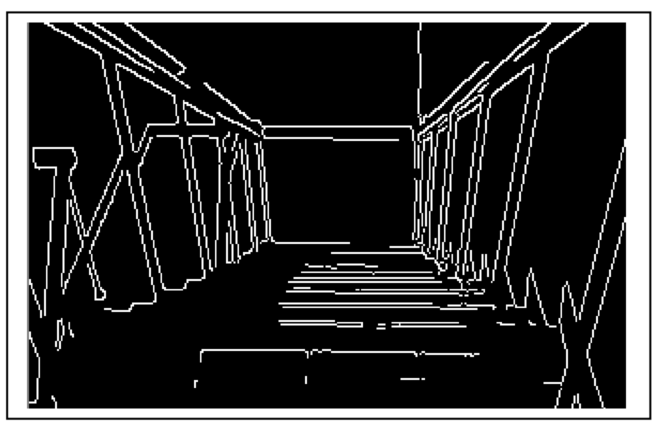


Fig. 11 Output of Canny Edge Detection

After the edge of the image is obtained, the Haar-like feature method will be used. This method is an algorithm for detecting obstacle objects. The basis for using this method is the comparison of real-time images with training images. Obstacle objects are successfully detected if a red square appears in the image. Fig. 12 shows obstacle objects in the image that have been recognized. The red rectangle is the region of interest (ROI) of the image that shows the position of the obstacle object.

Fig. 12 ROI Obstacle Object

When the ROI of the image is known, the coordinates of the obstacle location can be calculated. If the y-axis pixel coordinate value is greater than 120, it means that the obstacle object is 20 cm in front of the MR-CIAT so the Mini PC will send the 'Silent' command which will interrupt the movement information of the Covid-19 MR-CIAT Robot given by Arduino to the Motor Driver. The greater the value of the y-axis coordinates, it means that the Covid-19 MR-CIAT Robot is getting closer to the obstacle. Fig. 13 is the image coordinates provided by the Mini PC.

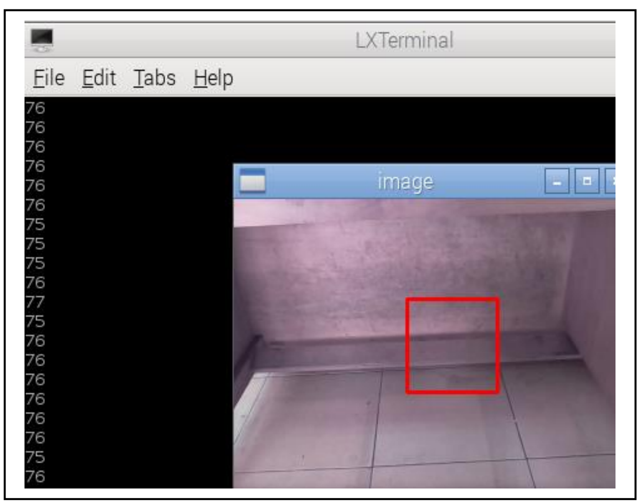


Fig. 13 Image Coordinates

B. Learning Process

The learning process on the machine aims so that the machine can recognize the object that it wants to detect. The image used consists of 54 samples of positive images in the form of stairs down and 27 samples of negative images. The image obtained is an image in RGB color format with a resolution of 360 x 240 pixels with a 24-bit bitmap file format for positive image samples. As for the negative image sample, the JPEG file format is used with color conversion to grayscale with a resolution of 360 x 240 pixels. Fig. 14 and Fig. 15 respectively are samples of positive and negative images used for the learning process.

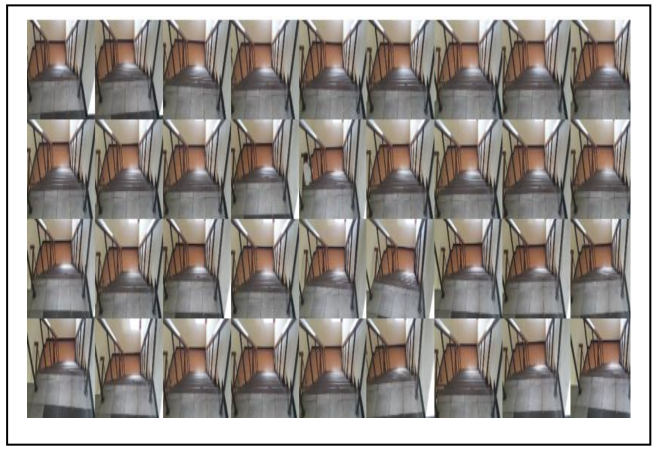


Fig. 14 Positive Image Sample

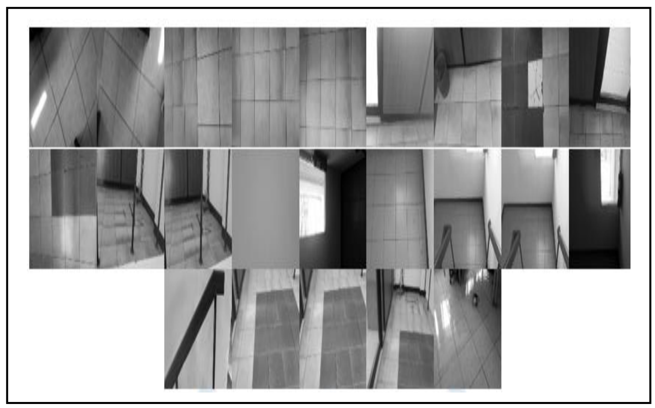


Fig. 15 Negative Image Sample

IV. Test Result

The light intensity tolerance test aims to find out how much the light intensity value can still be accepted by the camera to be able to recognize the stairs going down properly. In this experiment, 6 samples of detection object data were used in the form of a ladder with a top-down perspective and a distance between the table and the camera as far as 40 cm. There are 54 positive image training data with a scale factor parameter value of 1.1 and a minNeighbors of 7. Table 1 shows the observational data on the effect of changes in light intensity on the process of detecting objects descending stairs.

ISSN: 1942-9703 / CC BY-NC-ND

TABLE I
OBSERVATION DATA OF LIGHT INTENSITY ON DOWN STAIR OBJECTS

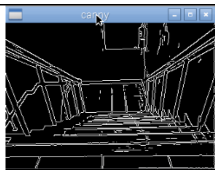
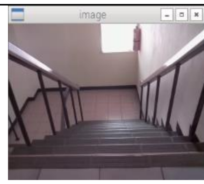
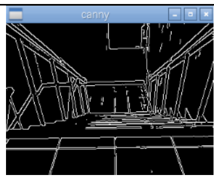
Data	Edge Image	Detection Result	Light Intensit y
1			15 (✕)
2			58 (✓)

TABLE II
DISTANCE OBSERVATION DATA

Data	Edge Image	Light Intensity	Distance (cm)	
			Nearest	Longest
1		58	75	94
2		74	75	89

Object detection testing using a camera is carried out to determine the success of the stairs-down detection with an image processing process that has been designed. The data to be processed is the closest and furthest distance that can still be received by the camera to recognize the stairs down. The experiment will be carried out 4 times with different environmental light intensity values. Table 2 shows the results of observational data in the form of maximum and minimum table detection distances.

Object detection testing using a camera aims to determine the success of the image processing process that has been designed and can determine the threshold position of the object on the camera. Table 3 shows the test results to determine the object detection threshold. This experiment was carried out 20 times with different camera angle positions.

Observation data of Raspicam camera sensors get the effect of changes in distance to the coordinates of object detection. The results of processing the observational data in Table 3 using linear regression get a regression equation of $Y=133.448 - 1.054X$ where the x-axis is the distance between the Covid-19 MR-CIAT Robot and the detected object and the y-axis is the object's detection position.

TABLE II
OBJECT DETECTION TEST

Data	Detect Position (Pixel)	Real Distance (cm)	Linear Regression Distance (cm)	% Error
1	11	121	121.855	0.701 %
2	18	115	114.479	0.455 %
3	22	110	110.265	0.240 %
4	26	105	106.051	0.991 %
5	31	100	100.783	0.776 %
6	36	95	95.515	0.539 %
7	41	90	90.246	0.273 %
8	48	85	82.871	2.569 %
9	52	80	78.657	1.701 %
10	56	75	74.442	0.749 %
11	61	70	69.174	1.194 %
12	66	65	63.906	1.712 %
13	69	60	60.745	1.226 %
14	75	55	54.423	1.059 %
15	78	50	51.262	2.463 %
16	85	45	43.887	2.536 %
17	90	37	38.619	4.192 %
18	99	30	29.136	2.964 %
19	101	25	27.029	7.504 %
20	110	20	19.654	1.762 %

The results of processing observational data show that the percentage of error in reading the coordinates of the object is not greater than ±7.5% and the reference value of the distance to be applied to the Covid-19 MR-CIAT Robot is 20 cm, so with the equation of the line, object coordinate reading value is 110. Obstacle detection was successfully implemented in indoor conditions with 51 positive image training data, 35 negative images and a scale factor parameter of 1.1, and a minNeighbor of 7.

V. CONCLUSION

Based on the observations and analysis, it can be concluded that the detection process is able to recognize obstacle objects in the light intensity range of 32 LUX to 103 LUX for table objects and the light intensity range of 58 LUX to 157 LUX for descending stairs objects. The medical robot – controlled assistive technology can detect both obstacles simultaneously in a light intensity range of 58 LUX to 103 LUX. Obstacle object detection distance is not affected by light intensity but on the stage formed in the cascade classifier process. The scale factor parameter affects the sliding window detection. The smaller the scale factor value, the more accurate the sliding window detection will be.

ACKNOWLEDGEMENT

Thanks to the Indonesian Ministry of Education and Culture Research and Technology. This research was funded by the Indonesian Collaborative Research Program Grant (PPKI) in 2021. The PPKI Grant Program is carried out by collaborative research between UNPAD, UPI, and ITS.

REFERENCES

[1] X. Guobo and W. Lu, "Image Edge Detection Based on OpenCV," *International Journal of Electronic and Electrical Engineering*, vol. 1, no. 2, 2013.

[2] N. S. Syafei, et. al., "Medical Robot Covid Orientation using IMU with PID Controller," *Internetworking Indonesia Journal*, vol. 13, no. 1, pp. 51-55, 2021.

[3] A. Turnip, T. Hidayat, and D. E. Kusumandari, "Development of brain-controlled wheelchair supported by raspicam image processing based Raspberry pi," In *2017 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT)*, pp. 7-11, IEEE, 2017.

[4] A. Turnip, A. I. Simbolon, M. F. Amri, and M. A. Suhendra, "Utilization of EEG-SSVEP method and ANFIS classifier for controlling electronic wheelchair," In *2015 International Conference on Technology, Informatics, Management, Engineering & Environment (TIME-E)*, pp. 143-146, IEEE, 2015.

[5] A. Turnip, D. Soetraprawata, M. Turnip, and E. Joelianto, "EEG-based brain-controlled wheelchair with four different stimuli frequencies," *Internetworking Indonesia Journal*, vol. 8, no. 1, 65-69, 2016.

[6] E. Joelianto, A. Turnip, and A. Widyotriatmo, *Cyber Physical, Computer and Automation System*, Springer Singapore, 2021.

[7] W. Xu, E. J. Lee, "Eye Detection and Tracking Using Rectangle Features and Integrated Eye Tracker by Web Camera*," International Journal of Multimedia and Ubiquitous Engineering*, vol. 8, no. 4, 2013.

[8] Nidhi, "Image Processing and Object Detection*," International Journal of Applied Research*, vol. 1, no. 9, pp. 396-399, 2015.

[9] H. Peng, "Application Research on Face Detection Technology Based on OpenCV in Mobile Augmented Reality," *International Journal of Signal Processing, Image Processing, and Pattern Recognition*, vol. 8, no. 2, pp. 249-256, 2015.

[10] S. Deswal, S. Gupta, B. Bhushan, "A Survey of Various Bilateral Filtering Techniques," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 8, no. 3, pp. 105-120, 2015.

[11] L. He S. Huang, "Modified Firefly Algorithm Based Multilevel Thresholding for Color Image Segmentation," *Neurocomputing Journal,* vol. 4, No 5, 2017.

[12] OpenCV Dev Team, *Basic Thresholding Operation*. https://docs.opencv.org/doc/tutorials/iimgproc/threshold/threshold.html. [Accessed on 16 May 2017]

[13] S. Vijayarani and M. Vinupriya, "Performance Analysis of Canny and Sobel Edge Detection Algorithms in Image Mining," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 8, 2013.

[14] M. H. Shokhan, "An Efficient Approach for Improving Canny Edge Detection Algorithm," *International Journal of Advances in Engineering & Technology*, vol. 7, no. 1, pp. 59-65, 2014.

[15] P. Menezes, J. C. Barreto, and J. M. Dias, "Face Tracking Based on Haar-Like Features and Eigenfaces," *IFAC Proceedings*, vol. 37, no. 8, pp 304-309, 2004.