International Journal of Information System & Technology
Akreditasi No. 36/E/KPT/2019 | Vol. 3, No. 2, (2020), pp. 140-151

Prototype of Blind Aid Tool With The Haar Like Feature

Ade Bastian[1], Dony Susandi[2], Usup Suparma[3]

[1,2,3]Teknik Informatika, Universitas Majalengka

bastiandicaprio@gmail.com[1], ds_77@gmail.com[2], usup.suparma7@gmail.com[3]

Abstract

Every human being is born perfectly but not all humans are born perfect, some of us are born have a deficiency in blindness. Blind people are someone whose vision is impaired from birth or because of an illness that causes their vision to be disrupted. Blind people need tools for their daily activities usually in carrying out blind activities using tools as a tool. Tools that are now widely circulating only as a detector can not yet identify objects, for that blind people neeed a tool that can identify the object. To make this happen, a prototype design research is done. Blind assitive devices use the haar like feature and extreme contour methods. Prototyping method is chosen so that users can play an active role during the process of designing, testing and implementing tolls. Contour is a curve or image that has similarities with objects or objects with functions using OpenCv. The conclusion of the detection results using the haar like feature method is the form of a string then processed using pygame so that the blind are able to know the object in fron of it.

Keywords: *Haar Like Feature, Prototype, Extreme Contour, Blind Aid Tool*

1. Introduction

Humans are creatures who have the desire to unite with each other and the natural environment around them. By using thoughts, instincts, feelings, desires and so on, humans react and interact with their environment. Humans interact with the environment using various ways one of which humans use the eyes as a vision to interact. Blindness is an individual who in one of his five senses is not functioning properly, that is, his sense of vision does not work properly so that the reception of daily information does not work like normal people in general. A blind person has the limitation to do his daily activities. [1]. A common tool used by blind people in Indonesia is blind sticks, while in many western countries it uses guide dogs to help move and increase security and independence when walking. Having enough information on the path of travel that will be passed by blind people can be more comfortable to navigate in an unknown environment. The stick is usually used when walking outside the room, but if the stick is not used inside the room for fear of damaging glassware. Another disadvantage of sticks is usually only to feel objects or obstacles that are under, and obstacles such as humans, motorbikes and cars are often not detected by assistive devices. Persons with visual impairments also experience problems in determining the distance of objects around them, for example friends whom they talk to. The shortage of sticks cannot yet define what objects are around, so from previous research blind sticks were limited to detecting objects, unable to define what objects prevent blind people from doing their daily activities. So in this study the author will do how to define objects that block blind people.

Object detection is an important element of various fields of computer vision. The basic goal is to find predetermined class objects in static images or video frames. Sometimes this task can be handled by extracting certain image features, such as edges, color areas, textures, contours, etc.[2]. A moving image or video has a variety of sizes and can be detected specifically. The shape of the animal, the shape of the house, the shape of the face can be detected by a surveillance camera as a scanning tool. The response time

ISSN: 2580-7250 IJISTECH

from the scanning process of these objects is also an additional requirement in the system. [3]. The camera used will be designed to detect humans and so on. The design of this tool is assisted by using a digital image processor with raspberry pi 3 model b. The software used uses OpenCV 3.4 with the Python 3.5 programming language. The image of the obstacle faced by the blind is captured by the webcam camera connected to the Raspberry Pi 3 model b then the image results are processed using the haar like feature algorithm until the shape or dimension has been known and the results of this detector are processed and produce sound output, so that blind people know what objects in front of him, so that he can respond to what must be done for example when there is a ladder in front of him, a priest must be prepared to lift his foot in step.

2. Research Methodology

Haar features, also known as Haar-like features, are a simple and inexpensive image feature based on intensity differences between rectangle-based regions that share similar shape to the Haar like wavelet. Haar features are used in object recognition as they feature digital image. They owe their name to the similarity with Haar wavelets. Viola and Jones joined their hand and found out the idea of using Haar wavelets and developed the method named Haar-like features [4]. There are 4 main concepts in detecting objects in images, namely [5] :

a) Training data using two models, such as the following:
   1) Positive sample, contains an image of the object you want to detect;
   2) Negative sample, contains an image other than the object you want to detect.
b) A rectangular feature called the Haar feature like feature is a feature based on a single wave longitude Sangkat consisting of one high interval and one low interval. In two-dimensional images, it means consisting of one light and one dark. So, every haar-like feature consists of a combination of black and white boxes.

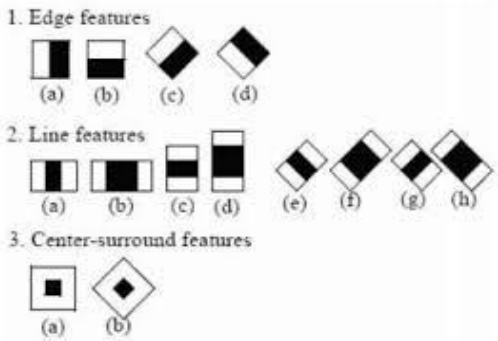


Figure 1. Fiture Haar Variation

Haar-like Feature value is obtained from the difference in the number of pixel values in the area with the number of pixel values in bright areas, can be seen in the following equation 2.1 :

$$F(Haar) = \sum Fwhite - \text{FBlack} \quad (1)$$

Where:

F(haar) is the total feature value,
ΣFwhite is a feature value in bright areas, and
ΣFblack is a feature value in a dark area.

c) Integral of images to detect features quickly, then neutralized (summed) based on pixel values contained in the image.

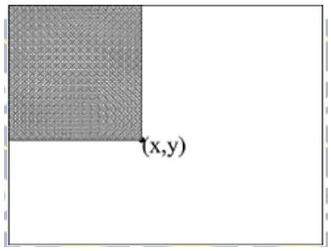


Figure 2. Image Integral (x,y)

d) Multi-level classifiers to connect many features more efficiently.

2.1. Contour

Contour is a list that contains points that can be said to represent a curva from an image. This picture can vary depending on the situation at hand. There are many ways to represent a curva in an image. Contour is described in OpenCV as a sequence (sequence) of information encoded about the location of the next point in the curve. The function that is in OpenCV, calculates contour from binary images. Binary images can be generated from a threshold that has an implicit angle as the boundary between positive and negative areas. [6]

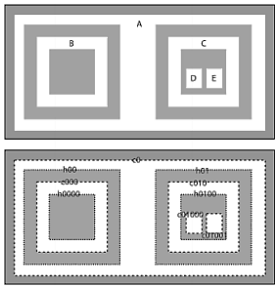


Figure 3. Contour on OpenCV

3. Results and Discussion

3.1. The Haar Like Feature Analysis

The analysis of processing image or object recognition is divided into three, namely in the first stage of the scaling process, the second stage of the grayscaling process, for the last stage is the trasholding stage. For the process flow in the Haar like feature can be seen in Figure 4 below :

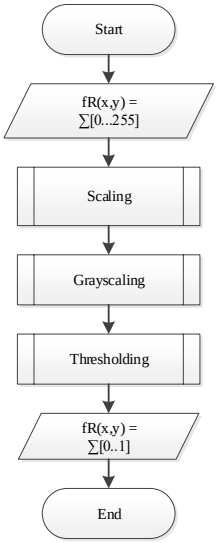


Figure 4. Process Flow on the Like Feature

a) Scaling stage
Digital images taken in realtime by the webcam will be minimized by using the interpolation method. This method uses the average value of a region to represent the region. The pixel value on the coordinates of the interpolated image is obtained by calculating the average value of 4 pixel values of the original image.

b) Grayscaling stage
The representation of the value of RGB (Red, Green, Blue) is converted into an image consisting of white and a black gradient called grayscale. To change RGB to greyscale the following formula can be used :

$$grayscale = 0.299R + 0.587G + 0.114B \quad (2)$$

Figure 5 is an image of grayscaling results :

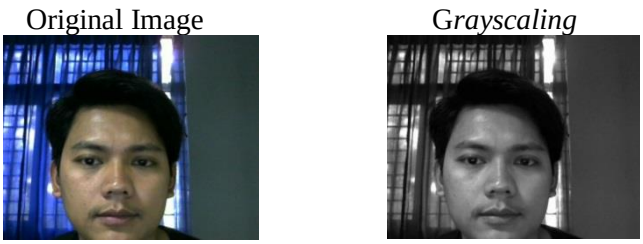


Figure 5. The Process of Changing RGB images into Grayscale image

1. Stage of thresholding
Next is the theresholding stage which is used to convert greyscale images to binary images. Threshold value is calculated by dividing the value of the greyscale result in the previous stage with the number of gray degrees (0 - 256) divided by 256 (desired gray degree value). The calculation process for thereshold values is as follows:

$$X = \frac{w}{b} \quad (3)$$

Where :
X = threshold comparison value
w = value of greyscaling results
b = 256 / a (a = 256)

To change the RGB image into a binary image, use the following rules:
1. If the image pixel value is ≥ x then the pixel value is 1
2. If the image pixel value is ≤ x then the pixel value becomes 0

Figure 6 is an image that is converted to a grayscale image into a binary image :

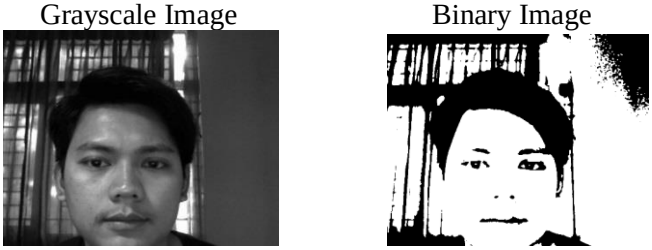


Figure 6. The Process o Changing the Grayscale image to Binary image

These three stages are the stages of object recognition using the haar like feature method on the design of the prototype for blind visual aids using the haar like feature method based on Raspberry Pi. These three stages are built using the library in OpenCv. OpenCv has facilitated this process so that it can easily be realized in the blind tool prototype.

3.2. Shape Detect Contours Analysis

The Shape Detect Contours analysis process is divided into six, namely in the first stage of the process of reading images (imread), the second stage is grayscaling, for the third stage is the stage of blur, for the fourth stage is thresholding, the fifth stage is the find contours and the last is to hold draw contours. For the process flow in Shape Detect Contours can be seen in Figure 7 below :

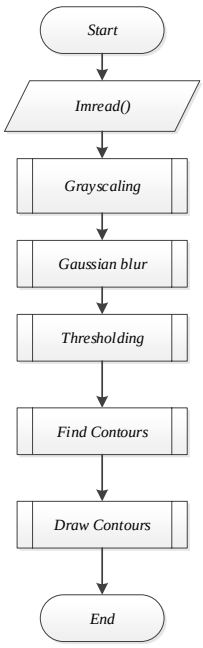


Figure 7. Flowchart Shape Detect Contours

Process analysis on shape detect contours is not much different from the stages in the haar like feature method, namely

a) Image reading (imread) is the process of opening an image on a disk
b) The grayscaling stage, which is the value representation of RGB (Red, Green, Blue), is converted into an image consisting of white and a black gradient called grayscale. For the formula as in formula 3.1.
c) The gaussian blur stage is the process of making an image that has been made white and black gradient to be slightly blurry (not too focused).
d) The thresholding stage is the process used to convert the gaussian blur image into a binary image.
e) The step of finding contours is the process of finding contours in image thresholding, then taking the biggest one.
f) The stage of draw contours is the process of drawing an outline of an object, then drawing each extreme point, where the far left is red, the right is blue, the top and bottom are teal colored.

3.3. Software Design

Process In designing this software will be discussed about the software that will be made on the prototype of blind visual aids using the haar like feature method with python programming.

a) Software Design on Arduino Nano.
The following is an image of software design on Arduino Nano in Figure 8 :

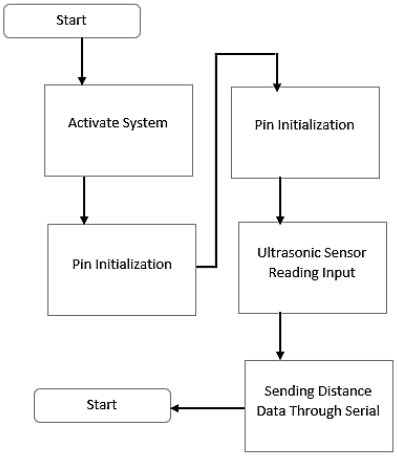


Figure 8. Software Design on Arduino Nano

For the steps in the flowchart of Figure 8 above, the algorithm notation is divided into three parts, namely the name of the program, the declaration and the Algorithm itself as below :

```
PROGRAM distance
        {Distance counting}
        procedure trig () {duration = (duration / 29) / 2}
        Declaration
        pTrig, pEcho, duration, distance: integer
        pTrig: OUTPUT
        pEcho: INPUT
        Algorithm:
        Distance = trig ()
        Write (distance)
```

b) Compounding Software in Python.

Software design in Python This design contains software design that can detect humans or other objects automatically by using the haar like feature method. The language used to create the program is the python programming language and is supported by OpenCv library for processing images taken by webcam cameras on raspberry pi. Python runs on raspberry instead of a personal computer (PC) or laptop as shown below :

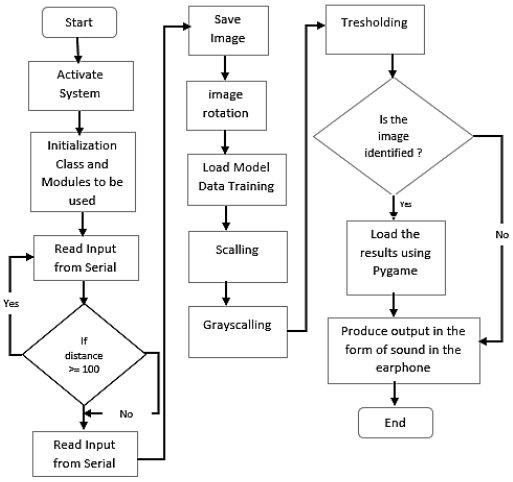


Figure 9. Software Design Flowchart

To better understand the flowchart in figure 9 above can be seen in the algorithm notation below which is divided into three parts, namely the program name, the variable name declaration used and the last algorithm itself :

```
PROGRAM Haar Like
{identification of objects using haarlike feature and extreme contour}
Time, string, argparse, cv2, playsound, serial, numpy, picamer: import

Class ObjectDetection (self, image)
{take pictures using the camera, save the picture, rotate the image in a
clockwise direction, call the training model, do the scaling, do the
grayscalling stage, continue the thresholding stage, and practice the
data with the camera results, match the camera results data with training
data}
       If (image == train) then
       Loadpygame = image
       endif

       Declaration
         np: numpy
         camera: PiCamera ()
         rawCapture = integer
       Algorithm:
         While I <true do
         Read (serial)
         If serial <= 100 then
         Print (distance)
         Read (camera)
         objectDetection (camera)
         endif
       endwhile
```

c) Composition of software on Raspberry Pi.

The software design on raspberry is in the form of operating system installation design that will be used on Raspberry Pi. The Raspberry Pi compounding scheme is as shown in Figure 10 below :

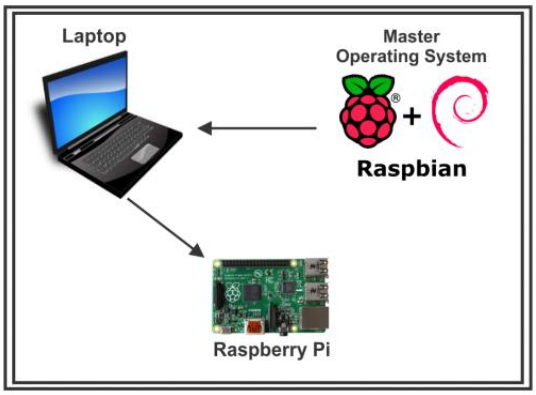


Figure 10. Raspberry Pi Installation Scheme

Figure 10 is a scheme for designing an operating system on raspberry pi. For the installation of the operating system, there are several components that must exist, namely the master operating system or operating system software that will be installed on Raspberry Pi, namely Raspbian (Raspbery Debian) which is a family of Linux operating systems specifically for Raspberry. Laptop or PC for installation and raspberry. Installation will be done on a laptop or PC that has been connected using a cable with raspberry pi. For data storage media on Raspberry Pi use a micro SD with a minimum capacity of 4GB. so the installation will be stored in the micro SD.

3.4. Making Hardware Prototype

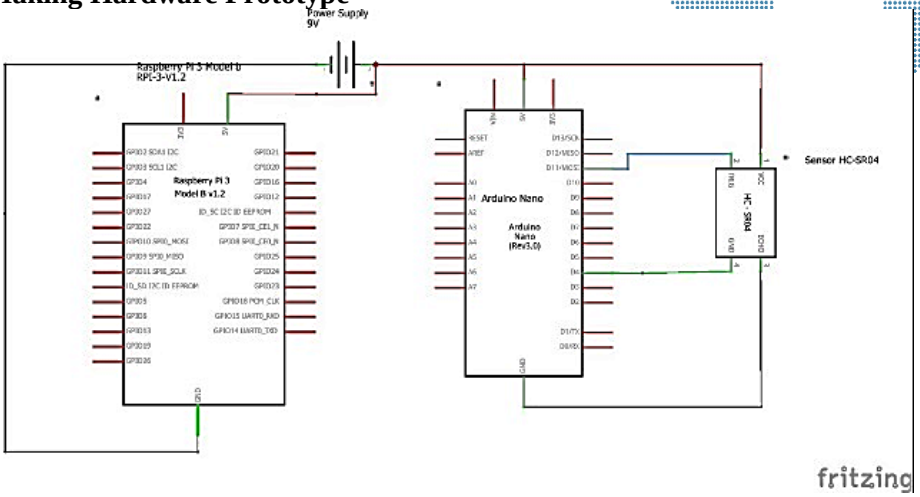


Figure 11. Prototype Electronic Circuit Tool

Explanation of figure 11 of the electronic prototype circuit of visual aids is as follows :

a) Prototype of this visual aids tool using Raspberry Pi 3 Model B with 5-12volt electrical power according to the specifications and needs of Raspberry Pi 3 Model b. Raspberry pi can use a power bank
b) The hardware used in the prototype of the visual impairment device is the HC-SR04 ultasonic sensor connected to Arduino Nano and then Arduino Nano is connected to the Raspberry Pi 3 Model B as a controller using the USB cable that is on the Raspberry.
c) Each hardware used consists of a minimum of Ground / GNG and VCC. For certain hardware VCC can be connected with the VCC pin on the Arduino Nano while the GND is still connected to the GND that is on the Arduino Nano.
d) The hardware that is in the prototype The visual aid consists of an input that is an ultrasonic sensor that has 4 pins that are connected to the GND pin, 5.5 volts, Digital Pin no. 4 and digital pin no. 11 contained on Arudino Nano.
e) The next input device is the PiCamera camera that is connected to the camera module port that is on the Raspberry Pi 3 model b.
f) After the input equipment there is a processing device that is using Raspberry Pi 3 model b.
g) For the output device consists of earphones connected to the 3.5mm port contained in the Raspberry Pi 3 model b.

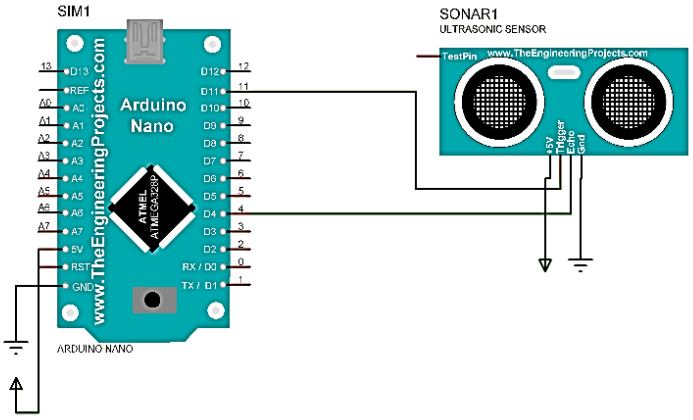


Figure 12. The Result of the Prototype Hardware Circuit

3.5. Software Testing

Testing this software is to find errors or shortcomings in the design of prototypes Blind assistive devices use the haar like feature method based on Raspberry Pi. This test consists of object identification testing using the haar like feature method. The identification of this object is to find out the accuracy of the haar like feature method in the python and OpenCV programs in detecting objects identified by ultrasonic sensors and taken by webcam cameras on raspberry pi. This test is carried out by detecting the test object can be seen in the table below:

a) Human Identification

Table 1. Human Identfication

No	Data Result from Camera Identification	Data Identified
1		Detected humans with an accuracy rate of 99.98%
2		Humans are detected with an accuracy of 99.85%

b) Motorcycle Identification

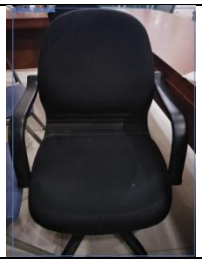
Table 2. Motorcycle Identfication

No	Data Result from Camera Identification	Data Identified
1		The object of the motorcycle was successfully detected with an accuracy rate of 99.53%
2		The object of the motorcycle was successfully detected with an accuracy rate of 99.96%

c) Chair Identification

Table 3. Chair Identfication

No	Data Result from Camera Identification	Data Identified
1		From the results of camera detection there are seats detected with an accuracy rate of 99.86%

No	Data Result from Camera Identification	Data Identified
2		From the results of camera detection there are detected seats with an accuracy rate of 99.99%

d) Car Identification

Table 4. Car Identfication

No	Data Result from Camera Identification	Data Identified
1		From the results of camera detection there are detected cars with an accuracy rate of 99.01%
2		From the results of camera detection there are cars detected with a level of accuracy of 100% for medium type cars while for pick up type cars detected with an accuracy rate of 91.33%

Hole Identification

Table 5. Hole Identfication

No	Data Result from Camera Identification	Data Identified	
		Yes	No
1		√	
2		√	

3.6. Overall System Testing

This test is done by turning on the appliance and starting counting with the stopwatch when it starts detecting obstacles until the identification process is complete and so on. The following are the results of detection and identification tests.

Table 6. Hole Identfication

No	Object Detected	Distance (cm)	Time (Seconds)
1	Chair	45	10
2	Human	47	7
3	Motorcycle	40	10
4	Car	50	12
5	Human	40	9
6	Chair	47	9

No	Object Detected	Distance (cm)	Time (Seconds)
7	Hole	47	11
8	Human	44	8
9	Motorcycle	48	10
10	Hole	48	9
11	Chair	90	8
12	Human	93	7
13	Hole	87	7
14	Motorcycle	94	8
15	Car	91	7
16	Human	92	7
17	Chair	93	7
18	Hole	94	7
19	Motorcycle	92	7
20	Human	90	7

From table 6 testing the response time of the tools above, the visual aids show that distance affects time with a distance between 40-50cm obtained an average of 8.6 seconds while for distances between 90-100 cm obtained an average time of 7.2 seconds.

The time obtained from the above experiments still did not get the expected time due to several factors that influenced the time including :

a) The processing speed on the Raspberry Pi 3 Model b is not as fast as the process that can be done on a laptop or on the Pesonal Computer (PC).
b) Light that is on the object affects the process of identifying the object itself if the light is too bright and too dark then the identification process takes longer.
c) The camera sensor used affects the quality of the captured image. In this study the PiCamera module camera uses a 5MP sensor resolution with a video resolution of 640 x 480.

4. Conclusion

Based on the results of the discussion and testing that has been carried out, the research conclusions are as follows :

a) Blind assistive devices use embedded systems, for embedded systems that are used that are using Raspberry pi 3 model b and Arduino Nano. There are two input devices that first use ultrasonic sensors as distance observers and the second uses a webcam camera for the shooting process.
b) The process of identifying objects first of all blind visual aids receive input from ultrasonic sensors then processed first using Arduino Nano after PiCamera Module takes the process of taking pictures, images are processed using the haar like classifier method found in the OpenCv library which has 3 stages namely Skaling, grayscaling and thresholding after all three stages have passed, the next step is matching the results of the thresholding with the training data images that have been trained before.
c) The result of processing the Haar like feature method is in the form of a string, then the results are processed using a python module called pygame so that the detection process can be heard by the blind through the media of earphone transmission.

Acknowledgments

We thank Universitas Majalengka for facilitating the research

References

[1] Asa, M. E. (2013). Implementasi sensor ultrasonik sebagai pemandu jalan bagi tunanetra berbasis mikrokontroler atmega8l. *Jurnal Teknologi Informasi* (p. 169). Yogyakarta: Jurnal Teknologi Infromasi.

[2] Mohammad Salah Udin and Afroza Yesmin Akhi. Horse Detection Using Haar Like Features. International Journal of Computer Theory and Engineering Vol. 8 No. 5 October 2016.
[3] Souhail Guennouni, Ali Ahaitouf and Anass Mansouri. A Comparative Study of Multiple Object Detection Using Haar-Like Feature Selection and Location Binary Patterns in Several Platforms. Hindawi Publishing Corporation Modelling and Simulation Engineering Volume 2015, Article ID 948960, 8 pages.
[4] Rafael Osorio, "Ball Detection via Machine Learning", Master‟s Thesis in Computer Science and Engineering Royal Institute of Technology Year 2009.
[5] Kusumanto, R. (2012). Aplikasi Sensor Vision untuk Deteksi Multiface dan Menghitung Jumlah Orang.
[6] Rosebrock, A. (2017). *Deep Learning For Computer Vision With Python.* chicago: pyimagesearch.com.

Authors

Ade Bastian, Ade Bastian is a permanent lecturer in the Informatics study program, Faculty of Engineering, Majalengka University. He completed his master's degree at STMIK LIKMI Bandung and has an interest in research in the field of multimedia informatics and computer vision.