

## P-D controller computer vision and robotics integration based for student's programming comprehension improvement

Nova Eka Budiyanta, Catherine Olivia Sereati, Lukas

Department of Electrical Engineering, Universitas Katolik Indonesia Atma Jaya, Indonesia

---

### Article Info

#### Article history:

Received Aug 13, 2019

Revised Jan 16, 2020

Accepted Feb 16, 2020

---

#### Keywords:

Computer Vision  
Integration  
P-D Controller  
Programming  
Robotics

---

### ABSTRACT

The 21st-century skills needed to face the speed of understanding technology. Such as critical thinking in computer vision and robotics literacy, any student is hampered by the programming that is considered complicated. This study aims at the improvement of student embedded system programming competency with computer vision and mobile robotics integration approach. This method is proposed to attract the students to learn about embedded system programming by delivering integration between computer vision and robotics using the P-D controller since both of the fields are closely related. In this paper, the researcher described computer vision programming to get the data of captured images through the camera stream and then delivered the data into an embedded system to make the decision of robot movement. The output of this study is the improvement of a student's ability to make an application to integrate a sensor system using a camera and the mobile robot running follow the line. The result of the test shows that the integration method between computer vision and robotics can improve the student's programming comprehension by 40%. Based on the Feasibility test survey, it can be interpreted that from the whole assessment after being converted to qualitative data, all aspects of the learning stages of programming application tested with the integration of computer vision and robotics fall into the very feasible category for used with a percentage of feasibility by 77.44%.

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



---

### Corresponding Author:

Nova Eka Budiyanta,  
Department of Electrical Engineering,  
Universitas Katolik Indonesia Atma Jaya,  
Kampus 3 BSD, Jl. Raya Cisauk Lapan, Sampora, Kec. Cisauk, Tangerang, Banten 15345, Indonesia.  
Email: nova.eka@atmajaya.ac.id

---

## 1. INTRODUCTION

Very rapid technological advances affect the speed of understanding technology. This is a challenge in the manufacturing sector in the industrial world which must meet consumers' needs consistently and with high quality [1]. To support this, in this modern era, it is very important to understand 21st-century skills which include computational thinking [2] as stated in 21st-century learning framework [3]. Based on the literature, programming competence strongly supports 21st-century skills. As applied to STT-PLN, to support this, STT-PLN has participated to improve student's comprehension of microcontrollers by creating practical modules and obtaining an increase in the value of respondents by 7.8% [4].

Robotic courses are often encountered in universities with major electrical engineering. Broadly speaking, the automation system in the industrial world is included in the electrical engineering students [5]. As in the Universitas Katolik Indonesia Atma Jaya, Robotics is taught as one of the subjects with a lot of interest. The purpose of this course is to create a robotic-based tool/technology. In general robotic lesson

does not only teach the students to become experts in the field of robotics but also to be able to develop essential competencies to be succeed in the real world [6].

Recently, Robotics has a big impact on the manufacturing industry [7]. Robotics is closely related to programming, and currently increasingly associated with the field of computer vision [8]. In the current era, Robotics is not far away from the application of artificial intelligence [9] as well in performing fast learning to obtain decision options based on multi-input multi-output problems [10]. The control system theory which is inseparable from mathematics is also applied to the results of this learning. Basically, there are a lot of enthusiasts to learn about robotics, but they are hampered by the programming that is considered complicated [11]. As a study conducted at the Department of School of Computer Science, University of Lincoln, UK, the method of integration between computer vision and robotics is very possible to be taught [12].

In addition to programming, the number of sensors that need to be implemented in robots also becomes a problem in assembling robots. This is because if there is one sensor that has a problem, it can cause a fatal error at the robot's rate. Similar to the line follower robot, it moves based on lines, so that there are more than one sensor needed to make the robot see the line below it [13-16].

For this reason, this integration method is proposed to encourage the focus of students learning about robot programming. In this study, the robot is implemented as a line follower robot equipped with a camera as a sensor, Raspberry Pi as an embedded system/processor, IC L293d as a motor driver, and 2 DC 12v motors as movers. We use Raspberry Pi as an embedded system that helped in data processing because it could be used effectively and smoothly for detection work [17-20]. We replace the photodiode sensor used for line follower robots to be a camera because in addition to the use of the cameras, it is easier than making sensor modules. Students can learn how to acquire images, and be processed in such a way that they can be data that is ready to be processed as a reference for moving motors. To support image processing, the OpenCV library is used in this study refers to many projects related to image processing that were also completed with OpenCV [21-24].

## 2. RESEARCH METHOD

Programming comprehension receive special attention and has been applied to the educational curriculum in many developed countries. However, programming has not received much attention by educational institutions in Indonesia. Although there are already a number of institutions that hold programming club activities, it is not enough as we should consider the benefits of programming to train student's critical thinking. Efforts are needed to further introduce programming to students. Therefore, this study proposed a method in which computer vision and robotics approaches are integrated to improve student's comprehension of programming. The integration steps can be seen in the following Figure 1.

Figure 1 shows that in order to increase the student's robot programming comprehension, the most important thing to do is doing study literature. At this stage, all knowledge about robot programming and computer vision is explored. Materials about programming are obtained by using reference books, surfing the internet, and presentation slides provided by the instructor. Broadly speaking, things explored include how the robot can recognize lines based on image acquisition from the camera, how to process data obtained from data acquisition from the camera, and how to use the image acquisition data that has been processed to be able to move the robot.

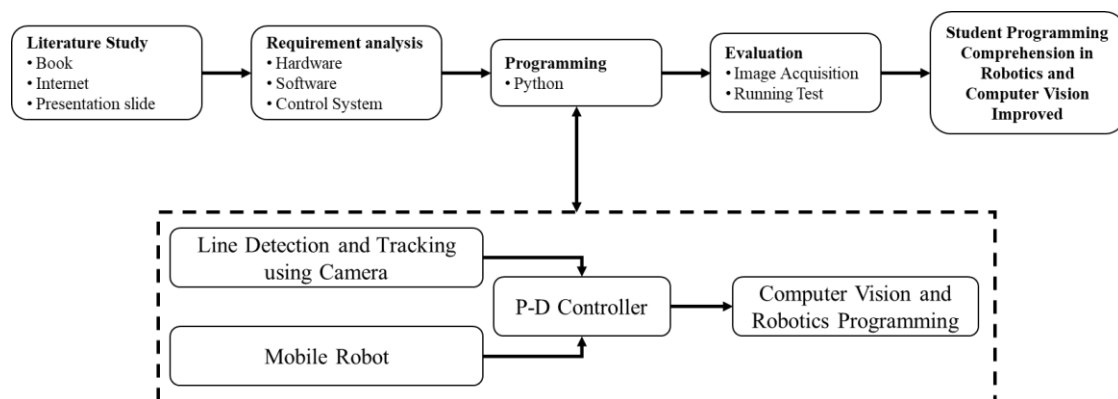


Figure 1. Computer vision and robotics integration programming methods

After a literature study has been carried out, the researcher started to analyze the needs. The needs analysis process is used to find out what hardware is used for image acquisition; embedded systems used to process data; hardware to make robots can move based on data that has been processed; and the hardware assembly structure of the robot. In addition to analyzing hardware requirements, the software is no less important to analyze, such as what software is used to program an embedded system in acquiring images from cameras, processing data acquisition, programming languages, and sending data to actuators to move a robot. In addition to hardware and software requirements, a control system is also needed to make the robot run stably in following the line. Then, the researcher implemented a control system together in data processing to produce mature data that is ready to be sent to the robot actuators.

After all aspects of the needs have been identified, then the process of assembling and programming robots begins. There are several choices of programming languages which are suitable for computer vision and robotics. One that is used in this study is the Python programming language.

After the assembly and programming process, the last is evaluation. There are 2 aspects of an evaluation carried out in this study. The first is image acquisition, it is used to test programs made by students for robots to acquire data from images captured by cameras. The second test is the robot running test, which is the robot runs on a white track with a predetermined black line.

To test the level of student's comprehension, the pretest-posttest technique was used in this study. There were 22 respondents involved in this study. The pretest was given at the beginning of the material to measure the extent of students comprehension of robot programming. After the achievement is done, the treatment begins using the method as stated in Figure 2. Finally, after completing the evaluation phase, a post-test was given to measure the extent of students comprehension after the learning process. After obtaining pretest and posttest data, an analysis was conducted to measure the level of change in students' understanding of robot programming with the integration of computer vision and robotics.

In addition to the Pretest-Posttest to test the student's programming comprehension improvement, data analysis techniques in this study were carried out using a quantitative descriptive approach. The quantitative descriptive approach was tested using descriptive statistics. The purpose of this analysis is to test the Feasibility of the integration of computer vision and robotics methods in helping to teach about programming. Quantitative data is in the form of calculation and/or measurement figures and can be processed by adding up and comparing with the expected number so that a percentage is obtained. So the feasibility percentage is determined by the following calculation:

$$\text{Percentage (\%)} = \frac{\text{Total Actual Score}}{\text{Total Ideal Score}} \times 100\% \quad (1)$$

The analysis of numerical data in this study uses a Likert scale. This is because the Likert scale is more reliable than the single team scale. After the percentage scale is formed, the data is interpreted into the Score Conversion table as in Table 1.

Table 1. Score Conversion

Percentage (%)	Criteria
100 – 76	Very Feasible
75 – 51	Feasible
50 – 26	Not Feasible
25 – 0	Very Infeasible

### 3. RESULTS AND ANALYSIS

#### 3.1. Hardware

The hardware needed for programming the integration of computer vision and robotics includes a camera. The camera is needed as a substitute for sensors to detect lines by acquiring path images to be represented in pixel coordinates in programming. The camera used in this study is Pi Camera V2 which can be used to take pictures with a resolution of 1080p and 720p, while for video can be recorded with a resolution of 640x480p. The embedded system device as a data processor used in this study is the Raspberry Pi 3 B + model that uses ARM (Advanced RISC Machine) as a processor and 1 GB of RAM. In addition to cameras and embedded systems, the robot is equipped with a motor driver module with IC L293d which is used to assist in regulating the speed and direction of rotation of two DC12v 450rpm motors on the robot. Then, to assemble all hardware components, a chassis made of acrylic boards with a thickness of 3mm is used in this study. In addition to the hardware from the robot itself, a PC/Laptop is needed to help the programming process by means of the Raspberry Pi remote system to the PC/Laptop.

**3.2. Software**

The softwares used in this study include the Operating System used on Raspberry Pi 3 B + models, applications for Raspberry Pi remotes using a PC using a WIFI connection, applications for making code, and libraries needed to program robot hardware. The operating system used in this study is Raspbian Stretch. To support the robot programming process, the VNC application is used to operate Raspberry Pi remotely through a laptop. Furthermore, the application used to create code on the Raspberry Pi is the default text editor from Raspbian. The last but not the least software is OpenCV library that was used to assist in making code to acquire image data captured by cameras.

**3.3. Programming**

In this phase, Python is used as a programming language to make robots function as line follower robots. There are 2 phases of programming carried out on this integration method.

**3.3.1. Image data acquisition**

Image data acquisition is used to process images captured by the camera and take data every pixel in the image. The purpose of this process is to produce the midpoint of the detected object, in this case a black line in the robot's path. The results of the programming process in image data acquisition can be seen in Figure 2.

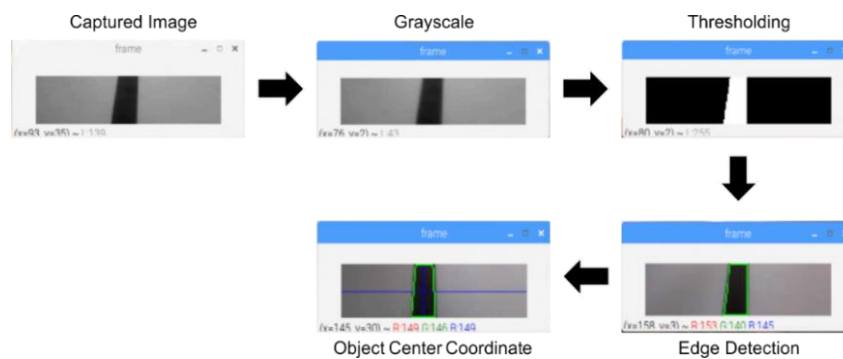


Figure 2. Image data acquisition

**3.3.2. Control system and data delivery to actuators**

After the center coordinate data of the object captured by the camera is obtained, the data is then used to be processed again using a control system to be a value to control the rotation, direction, and speed of the two actuators. The control system used in this study is the P-D controller. The stages of implementing data from the center object reading into the control system are as follows. The frame divided into 9 parts, each part has a weight value with the following details stated in Table 2. Which the weight value will be considered as an error value of the X value in the middle of the frame.

Table 2. The weight value determination of each frame

The Range of X (px)	Weigth Value
0 <x< 20	4
20<x<37,5	3
37,5<x<55	2
55<x<72,5	1
72,5<x<87,5	0
87,5<x<105	-1
105<x<122,5	-2
122,5<x<140	-3
140<x<160	-4

The next stage is the application of the weight value obtained from image processing into the control system. This study applies PD controller to support the control system. There are many studies that apply PD controllers to robots [25-26] and also develop PD controllers so that they are more robust [27]. The PD controller system parameter is based on the diagram which can be seen in Figure 3.

Based on the diagram block in Figure 3, an algorithm can be interpreted to regulate robot motion into pseudo code as follows:

```

setpoint      = 0
previous_error = 0
Kp           = (tuning)
Kd           = (tuning)
error        = setpoint - weight_value
derivative   = (error - previous_error)/dt
pd_out       = (Kp * error) + (Kd * derivative)
previous_error = error
wait(dt)
PWMLeft(constant_speed + pd_out)
PWMLRight(constant_speed - pd_out)
    
```

By using the algorithm above, the robot's logic can be interpreted as shown in flowchart Figure 4.

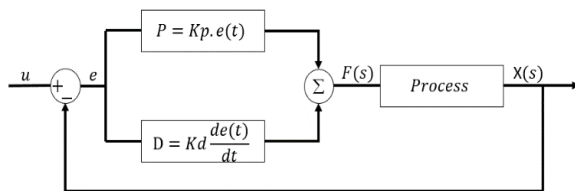


Figure 3. P-D controller diagram block

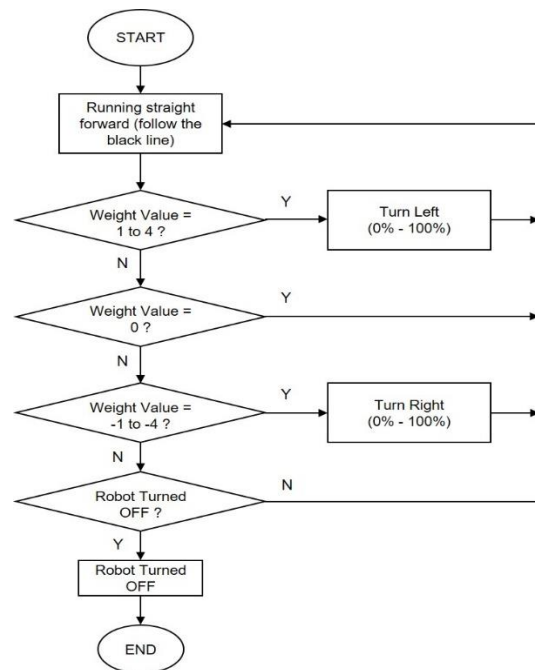


Figure 4. Flowchart system

**3.4. Evaluation**

**3.4.1. Robot testing result**

The results of computer vision and robotics programming integration testing can be seen in Figure 5.

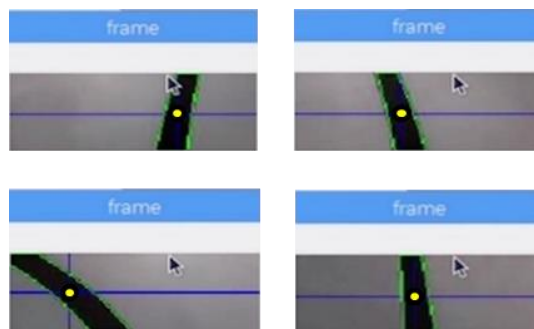


Figure 5. Mobile robot running test result

**3.4.2. The result of pretest posttest**

The results of student programming comprehension improvement are assessed based on the pretest and posttest questions done by students at the beginning and the end of the meeting. Pretest and posttest questions were done by students amounted to 20 items including material about Image Processing, Python programming, use of OpenCV, P-D Control System, and GPIO Programming on Raspberry Pi 3. Pretest and posttest questions were done for 50 minutes with the expectation of maximum results. Based on the data obtained, the average value of the pretest and posttest of all students has increased. An improvement in the average pretest and posttest scores can be seen in Figure 6.

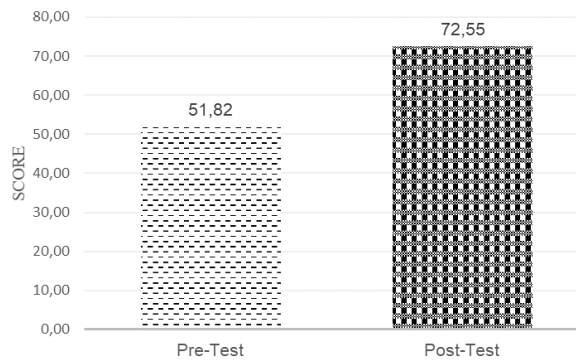


Figure 6. The improvement of average pre-test and post-test result

Student programming comprehension results improved by 40% from average Pre-Test score of 51.82 to average Post-Test score of 72.55. Based on this result, the learning methods of Computer Vision and Robotics integration are appropriate to be applied to the Robotics course.

**3.4.3. Feasibility test result**

The Feasibility test in the study is based on the usability aspect of applying the learning stages of programming with the integration of computer vision and robotics. Responses to usability aspect were collected using a questionnaire with 7 items representing sub-aspects of Understandability, Operability, and Learnability. Data on the results of the Feasibility test in this study can be seen in Table 3.

**Table 3. Feasibility test questionnaire results**

Aspect	Sub-aspects	Indicators	Avg. Point	Max. Avg. Point
Usability	Understandability	1	3.05	4
		2	3.27	4
		3	3.14	4
	Operability	4	2.95	4
		5	3.27	4
		6	2.82	4
	Learnability	7	3.18	4
<b>Total</b>			<b>21.68</b>	<b>28</b>

Table 3 shows that the total of average feasibility point is 21.68 instead of 28. According to the data, the feasibility percentage can be shown using this equation as stated before. The result of feasibility percentage calculation can be seen as follows:

$$Feasibility = \frac{Total\ Avg.\ Point}{Total\ Max.\ Avg.\ Point} \times 100\% \tag{2}$$

$$Feasibility = \frac{21.68}{28} \times 100\% \tag{3}$$

$$Feasibility = 77.44\% \tag{4}$$

Based on the Feasibility test survey, it can be interpreted that from the overall assessment after being converted to Qualitative Data, all aspects of the testing of the application of learning programming stages with the integration of computer vision and robotics are included in the Very Feasible category for use with a percentage of feasibility by 77.44%.

#### 4. CONCLUSION

The impact of applying the learning stages of programming with the integration of computer vision and robotics can improve student's programming comprehension in robotics. Student's comprehension improvement was observed based on pre-test and post-test results. The average value of student's understanding increased by 40% from the original average pre-test score was 51.82 to 72.55. In addition, based on the Feasibility test that has been analyzed, this study falls into the very Feasibility category for use with a percentage of Feasibility of 77.44%. The integration of Computer Vision and Robotic programming can still be developed for further research such as the introduction of objects on the camera, calculation of the object distance from the robot, and the object tracking system on the robot.

#### ACKNOWLEDGEMENTS

The authors thank the Electrical Engineering study program and the Faculty of Engineering at Universitas Katolik Indonesia Atma Jaya that has fully supported our research.

#### REFERENCES

- [1] Mohd RMS, Khalil AMA, "Synchronous mobile robots formation control," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 16, no. 3, pp. 1183-1192, June 2018.
- [2] Keser H., Uzunboylu H., Ozdamli F, "The trends in technology supported collaborative learning studies in 21st-century," *World Journal on Educational Technology*, vol. 3, no. 2, pp. 103-119, 2011.
- [3] Sezer K, "Importance of Coding Education and Robotic Applications for Achieving 21st-Century Skills in North Cyprus," *International Journal of Emerging Technology in Learning*, vol. 12, no. 1, 2017.
- [4] Indrianto, Mellia N., Rakhmat A., Riki R., "Embedded system practicum module for increase student comprehension of microcontroller," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 16, pp. 53-60, February 2018.
- [5] Ade GA., et al, "Low-cost and portable process control laboratory kit," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 16, no. 1, pp. 232-240, February 2018.
- [6] Kathia P., Belen C., Joaquin G., Vidal M., NXT, "Workshop: Constructionist Learning Experiences in Rural Areas," *Proceeding of Intl. Conf. on SIMPAR (Simulation, Modelling, and Programming for Autonomous Robots) Workshop "Teaching robotics, teaching with robotics"*, pp. 504-513, November 2010.
- [7] Ihsan AT., Hamzah MM, "Implementation of Controlled Robot for Fire Detection and Extinguish to Closed Areas Based on Arduino," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 16, no.2, pp. 654-664, April 2018.
- [8] G. Bebis, D. Egbert and M. Shah, "Review of computer vision education," in *IEEE Transactions on Education*, vol. 46, no. 1, pp. 2-21, February 2003.
- [9] Catherine O. S., et al, "Architecture design for a multi-sensor information fusion processor," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, pp. 362-369, 2019.
- [10] Karel O. B., et al, "Cognitive artificial-intelligence for doernenburg dissolved gas analysis interpretation," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 1, pp. 268-274, February 2019.
- [11] Yorah B., Marco AG, "Why is programming so difficult to learn?: Patterns of Difficulties Related to Programming Learning Mid-Stage," *ACM SIGSOFT Software Engineering Notes*, vol. 61, no. 6, November 2016.
- [12] G. Cielniak, N. Bellotto and T. Duckett, "Integrating Mobile Robotics and Vision With Undergraduate Computer Science," in *IEEE Transactions on Education*, vol. 56, no. 1, pp. 48-53, February 2013.
- [13] Islam M. S., Rahman A. M., "Design and Fabrication of Line Follower Robot," *Asian Journal of Applied Science and Engineering*, pp. 127-132, 2013.
- [14] Prananjali K., Vishnu A., "Sensor Based Black Line Follower Robot," *International Journal of Engineering Research & Technology (IJERT)*, vol. 3, September 2014.
- [15] Anupoju A. V., et al, "Design to Implementation of A Line Follower Robot Using 5 Sensors," *International Journal of Engineering and Information Systems (IJEAIS)*, vol. 3, pp. 42-47, January 2019.
- [16] Mehran P., Mehdi S. M., Mahdi R. G., "A Line Follower Robot from design to Implementation: Technical issues and problems," *Proceeding of the 2nd International Conference on Computer and Automation Engineering (ICCAE)*. Singapore, pp. 5-9, 2010.
- [17] Sumardi, Muhammad T., Munawar A, "Street mark detection using Raspberry PI for Self-driving System," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 16, pp. 629-634, 2018.
- [18] Onkar R., et al., "Object Detection on Raspberry Pi," *International Journal of Engineering Science and Computing*. vol. 7, no. 3, 2017.

- [19] Ali A. A., Sara A. R., "Computer vision for object recognition and tracking based on Raspberry Pi," *International Conference on Change, Innovation, Informatics, and Disruptive Technology ICCIIDT'16*. London, pp. 177-189, 2016.
- [20] Dhanashree V. M., Mrinal R. B., "Real Time Object Detection and Tracking using Raspberry Pi," *International Journal of Engineering Science and Computing*, vol. 7, no.6, 2017.
- [21] Grzegorz M., Przemyslaw M., "Line Following Robot Real-Time Virtebi Track-Before-Detect Algorithm," *Przegląd Elektrotechniczny*, pp. 71-74, 2017.
- [22] Andrey K., Vladislav M., "New algorithms for satellite data verification with and without the use of the imaged area vector data," *Proceeding of WSCG 23rd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pp. 1-7, 2015.
- [23] Guobo X., Wen L., "Image Edge Detection Based on Opencv," *International Journal of Electronics and Electrical Engineering*. vol. 1, no. 2, June 2013.
- [24] João G., David R., Filipe S., "Perspective correction of panoramic images created by parallel motion stitching," *Proceeding of WSCG 23rd International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pp. 125-131, 2015.
- [25] Salah M. S., et al., "Design and simulation of robotic Arm PD Controller Based on PSO," *University of Thi\_Qar Journal for Engineering Sciences*, pp. 18-24, 2019.
- [26] Adrian P., "The use of closed-loop control systems in Botball," *2018 European Conference on Educational Robotics*. Malta, 2018.
- [27] Hazem I. A., Ali H. S., "Robust PI-PD controller design for systems with parametric uncertainties," *Engineering and Technology Journal*, vol. 34, no. 11, 2016.

## BIOGRAPHIES OF AUTHORS



**Nova Eka Budiayanta** got his Bachelor Degree in Mechatronics Engineering Education from Universitas Negeri Yogyakarta, then pursuing his Master Degree in Electrical Engineering Education from Universitas Negeri Yogyakarta and Electrical Engineering Master Program from Universitas Katolik Indonesia Atma Jaya. He has experience in hardware-software programming for education. Now acting as lecturer and the Head of Embedded System Laboratory of Electrical Engineer Department, Universitas Katolik Indonesia Atma Jaya with concern in image processing, robotics, and machine learning research field.



**Catherine Olivia Sereati** is a lecturer and researcher at Universitas Katolik Indonesia Atma Jaya. Catherine's interest subject of researches are Electronic Instrumentation System and System on Chip (SoC). She was also involved in several research projects to design cognitive instrumentation systems. Some of them are a building a software cognitive interpretation of ship movements, for Indonesian marine security purposes, and Cognitive Electro Cardiograph (ECG) design. Currently her research project is focusing to designing the architecture of Cognitive processor.



**Lukas** got Bachelor degree of electrical engineering (EE) from ITB, then pursued Master of Artificial Intelligence and PhD in Electrical Engineering, both from KU Leuven, Belgium. Currently acting as the Head of Master of EE program Unika Atma Jaya, with research in image processing, natural language processing and apply cognitive engineering. Currently acting as the President of Alumni KU Leuven Chapter Indonesia, President of Indonesia AI Society, Secretary of Indonesia HoneyNet Project. His main interests are in artificial intelligence, natural language processing and computer security.