

DEEP LEARNING-BASED OBJECT RECOGNITION ROBOT CONTROL VIA WEB AND MOBILE USING AN INTERNET OF THINGS (IoT) CONNECTION

^aBasuki Rahmat, ^bBudi Nugroho

^{a,b} Informatics Department, Faculty of Computer Science, Universitas Pembangunan Nasional "Veteran" Jawa Timur, Jalan Raya Rungkut Madya, Gunung Anyar, Surabaya Jawa Timur 60294, Indonesia
E-mail: basukirahmat.if@upnjatim.ac.id, budinugroho.if@upnjatim.ac.id

Abstract

The paper presents the intelligent surveillance robotic control techniques via web and mobile via an Internet of Things (IoT) connection. The robot is equipped with a Kinect Xbox 360 camera and a Deep Learning algorithm for recognizing objects in front of it. The Deep Learning algorithm used is OpenCV's Deep Neural Network (DNN). The intelligent surveillance robot in this study was named BNU 4.0. The brain controlling this robot is the NodeMCU V3 microcontroller. Electronic board based on the ESP8266 chip. With this chip, NodeMCU V3 can connect to the cloud Internet of Things (IoT). Cloud IoT used in this research is cloudmqtt (<https://www.cloudmqtt.com>). With the Arduino program embedded in the NodeMCU V3 microcontroller, it can then run the robot control program via web and mobile. The mobile robot control program uses the Android MQTT IoT Application Panel.

Key words: *Surveillance, Robot, BNU 4.0, IoT, Deep Learning.*

INTRODUCTION

Intelligent Robot is a robot that is designed and created to solve certain problems intelligently using artificial intelligence algorithms. The use of intelligent robots for various purposes continues to grow and increasingly sophisticated, such as industrial robots and service robots. Industrial robots are often used to help complete the production process in factories or industries. While service robots are widely used to help medical services, household chores, restaurants, education, games, entertainment, and so on. Some examples of industrial robot applications include industrial hybrid spray-painting robots [1], industrial footwear [2], milling processes [3], peg-in-hole (PiH) assemblies [4], polishing systems [5], and many others. While some examples of the application of service robots include medical service robots [6], household service robots [7], restaurant service robots [8], education service robots [9], game and entertainment service robots [10], and many more other.

Of the many intelligent robots, one of which is immediately visible intelligence is a surveillance robot. The robot that uses the camera as its flagship sensor and of course is equipped with an object recognition algorithm. This surveillance robot is then used as an example of intelligent robot programming from this research. Some examples of the application of surveillance robots that have been carried out by previous researchers and the technology at a glance are described here.

One example of an android-based mobile robot for monitoring and supervision is as in the paper [11]. This surveillance robot is designed and made cost-effective by using an Arduino microcontroller along with a motor shield and an Android smartphone running an Operating System. The robot is equipped with a video camera and a Wifi robot link. The system utilizes Application Programming Interfaces (API) provided for the operating system. The robot can be controlled remotely using a Wifi module and a microcontroller, and a smartphone interface that is embedded in the robot. The camera on the robot is used to capture and record video in real-time from the robot. The robot can be controlled based on visual feedback from the same smartphone. Four-wheeled DC motor helps navigate robots

and ultrasonic sensors to avoid obstacles. The camera is attached to a Wifi robot link that allows it to capture the environment or object of concern. The results of trials with various obstacle positions show the flexibility of the robot to avoid it and have shown decent performance and obtained a communication range of nearly 50 meters, which is good enough for many surveillance applications.

Another example of a surveillance robot that is very challenging now is an unmanned vehicle or an autonomously driving car as shown in the paper [12]. This vehicle can be realized as advances in microelectronic technology, information, communication, materials, sensors, cyber-physical systems (CPS) and the Internet of Things (IoT). CPS and IoT are experiencing explosive growth associated with sophisticated cellular systems such as intelligent automotive and avionics systems, cellular robots, and wearable devices. Specifically, smart vehicles or cars involve large amounts of instant data from various complex sensors or other systems, and are required to provide long-term, autonomous services. To adequately meet these demands, sophisticated embedded computing and embedded design technologies are needed.

MATERIAL AND METHODS

The intelligent surveillance robot in this study was named BNU 4.0. BNU is short for Bela Negara University, dedication to Bela Negara Campus. The robot body is a modified Remote Control car (RC car). This robot control brain is a microcontroller NodeMCU V3. Electronic board based on ESP8266 chip. With this chip, NodeMCU V3 has the ability to connect to the Internet of Things (IoT) cloud. So that the robot can move forward, backward, turn right, and turn left, use the L298N motor driver. This robot is also equipped with a Kinect Xbox 360 camera. The design of the BNU 4.0 surveillance robot control system via web and mobile equipped with a Kinect Xbox 360 camera and a Deep Learning algorithm for object recognition in front of it, is shown in Figure 1.

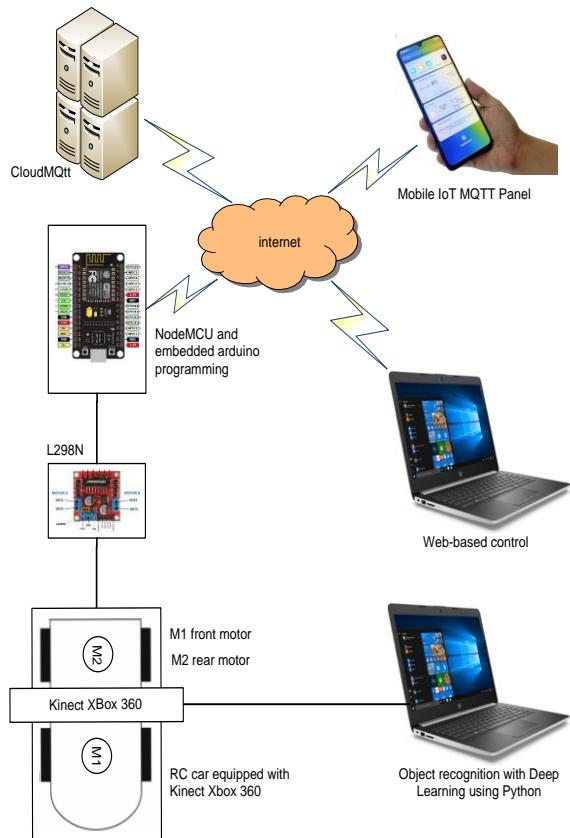


Fig 1. The design of the BNU 4.0 surveillance robot control system

Whereas the BNU 4.0 surveillance robot system block diagram as shown in Figure 2.

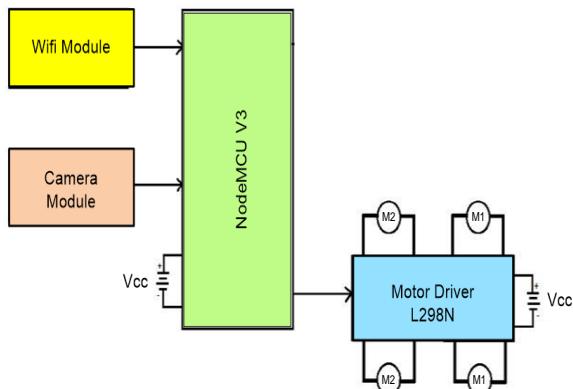


Fig 2. Intelligent robot system block diagram

From Figure 1 and Figure 2, it can be explained as follows. Robotic bodies in the form of RC cars that have been modified. RC cars usually have two motors, namely the front motor (M1) and rear motor (M2). The front motor is used to turn right and turn left, while the rear motor is used to move forward and backward. To run both of these motors, the L298N motor driver is used. The motor driver is connected to the NodeMCU V3 microcontroller with the circuit relationship as

in Figure 3. The NodeMCU V3 is based on the ESP8266 chip. With this chip, NodeMCU V3 has the ability to connect to the Internet of Things (IoT) cloud. The IoT cloud used in this research is the MQTT cloud (<https://www.cloudmqtt.com>). With the Arduino program embedded in the NodeMCU V3 microcontroller, a web-based robot control program can then be run. With the Arduino program designed and embedded in the NodeMCU V3 microcontroller, the robot control program can then be run through the Android Apps (IoT MQTT Panel). This robot is also equipped with a Kinect Xbox 360 camera. This camera is connected to a computer (laptop). With the Deep Learning program on the computer, the system will recognize the object in front of it.

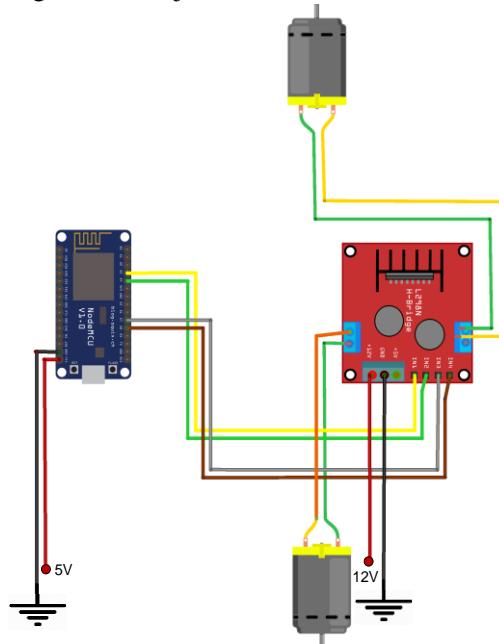


Fig 3. Electronic circuit intelligent robot system

From Figure 3, IN 1 and IN 2 from the L298N motor driver, connected to pins D3 and D4 of the NodeMCU V3 microcontroller. It is used to control the rear motor (M2), for forward and reverse motion. Whereas IN 3 and IN 4 from the L298N motor driver, are connected to pins D7 and D8 of the NodeMCU V3 microcontroller. It is used to control the front motor (M1), to turn right and turn left.

Meanwhile, the Deep Learning algorithm used for object recognition in front of the robot uses the Deep Neural Network (DNN) module from OpenCV. The steps for object recognition using DNN with OpenCV consist

of: import model from disk, initialize the video stream, loop over the frames from the video stream, grab the frame dimensions and convert it to a blob, pass the blob through the network until finally objects detection and recognition. Flowchart of object recognition using the Deep Learning algorithm with DNN from OpenCV, is shown in Figure 4.

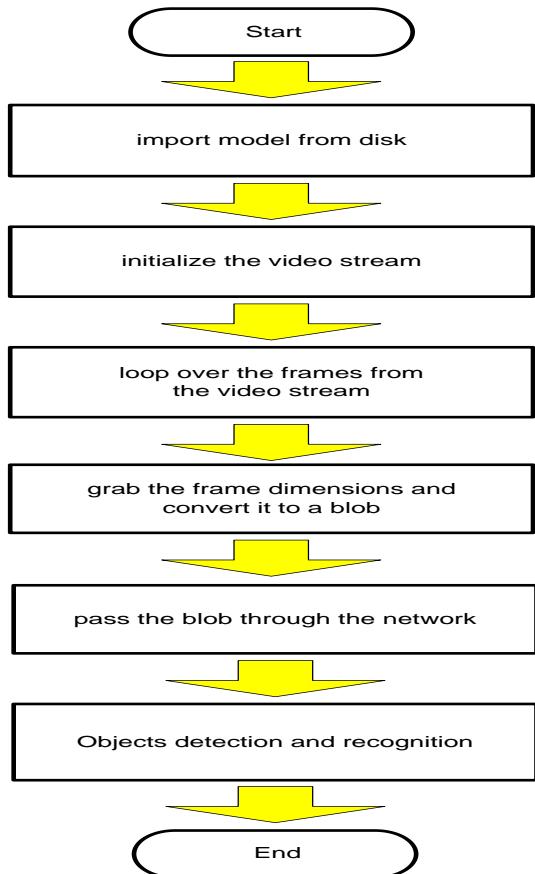


Fig 4. Flowchart of object recognition using the Deep Learning algorithm with DNN from OpenCV

RESULT AND DISCUSSION

As previously explained, that the BNU 4.0 Intelligent Robot body is a modified RC car. This RC car has two motors, namely the front motor (M1) and rear motor (M2). The front motor is used to turn right and turn left, while the rear motor is used to move forward and backward. To run both of these motors, the L298N motor driver is used. The motor driver is connected to the NodeMCU V3 microcontroller based on the ESP8266 chip. With this chip, NodeMCU V3 has the ability to connect to the IoT cloud. Thus, the Arduino program created will be used to control robots via the web and mobile via an internet

connection. For mobile control, use Android Apps (IoT MQTT Panel).

Arduino program for robot control, adapted to the electronic circuit of robots. Where IN 1 and IN 2 from the L298N motor driver, connected to pins D3 and D4 of the NodeMCU V3 microcontroller. It is used to control the rear motor (M2), for forward and reverse motion. Whereas IN 3 and IN 4 from the L298N motor driver, are connected to pins D7 and D8 of the NodeMCU V3 microcontroller. It is used to control the front motor (M1), to turn right and turn left. The following is an example of an arduino program script that is important for the basic control of this robot.

The Arduino program is tailored to control via mobile IoT. The use of Servers, Ports, etc. is adjusted with information from the MQTT Cloud server, see Figure 5. Likewise regarding the panels used in the IoT MQTT Panel, which consists of Onward, Turn left, Turn right and Backward panels. Filled with topics Onward, Turn left, Turn right and Backward. Each with payloads 1, 2, 3, and 4.

Topic is the name of the message, in which the client in this case the NodeMCU V3 microcontroller publishes, subscribes to, or does both for a topic. In this BNU 4.0 Intelligent Robot, the settings are in the Arduino program. So when the panel on the mobile via the IoT MQTT Panel application, the corresponding topic or panel is pressed, then clients who have subscribed to the topic can respond. In this case the client (Robot BNU 4.0 with NodeMCU V3 microcontroller) will move forward, turn left, turn right or backward.

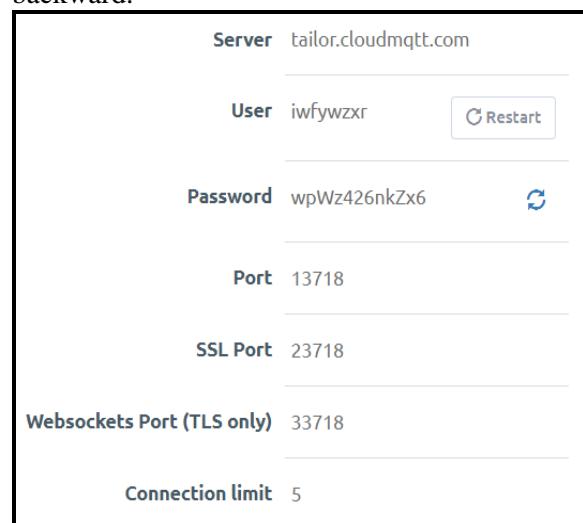
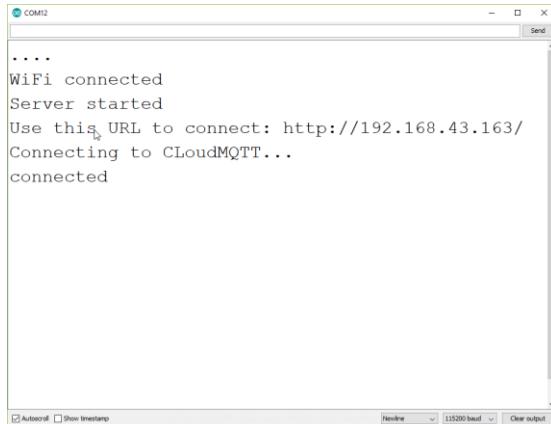


Fig 5. Detailed server information and others

For testing the Arduino program, it can be seen in the Tools - Serial Monitor menu. Then it will appear, the results on the serial monitor as shown in Figure 6.



....
WiFi connected
Server started
Use this URL to connect: http://192.168.43.163/
Connecting to CloudMQTT...
connected

Fig 6. Display on the monitor serial

Furthermore, for controlling BNU 4.0 Robot via the web, just run through the browser. According to the IP address obtained as shown on the serial monitor in Figure 6, we can type the IP address. In this example, the IP address <http://192.168.43.163> is obtained. After running in the browser the control buttons will appear to move the BNU 4.0 Robot to walk forward, backward, turn right and turn left. See Figure 7. Can be tried to suppress, and see the response to how the robot moves. The L298N motor driver battery is in a working condition.

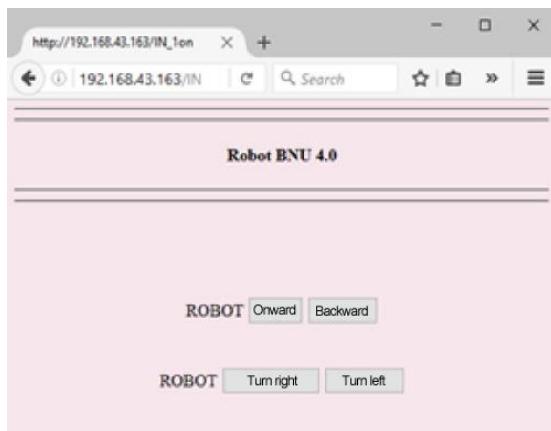


Fig 7. BNU 4.0 Robot Control via web

Then, for controlling BNU 4.0 Robot mobile, please run the IoT MQTT Panel on the mobile. If the robot has successfully connected to the Cloud MQTT, the control buttons will appear to move the BNU 4.0 Robot to walk forward, turn left, turn right and backward, through the cellphone. See Figure 8. Can be tried to suppress, and see the response to how

the robot moves. The L298N motor driver battery is in a working condition.

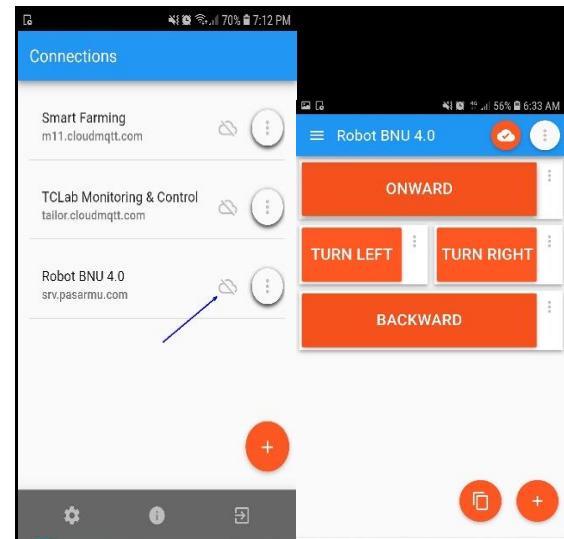


Fig 8. Connection to Cloud MQTT and BNU 4.0 Robot control panel via mobile

To see the response to the results of a button press on the cellphone, it can also be monitored via a serial monitor. Figure 9, shows an example display on a serial monitor if all four buttons have been pressed. And BNU 4.0 Robot moves as programmed.



....
WiFi connected
Server started
Use this URL to connect: http://192.168.43.163/
Connecting to CloudMQTT...
connected
Message received: Onward
payload: 1
Message received: Turn left
payload: 2
Message received: Turn right
payload: 3
Message received: Backward
payload: 4

Fig 9. Example monitoring of cellphone button presses through serial monitors

Furthermore, for real time object recognition in front of the robot, the Deep Learning algorithm is used by utilizing the DNN module from OpenCV. The DNN module in the library implements the forward pass using Deep Network, with previous training using Caffe. Some of the DNN module functions for deep learning programming purposes include: loading images from disk using the cv2.dnn.blobFromImage or cv2.dnn.blobFromImages functions. To be able to import models directly from various frameworks via the "create"

method, an example for this robot is: cv2.dnn.createCaffeImporter. If we only want to read the serial model from disk directly use the "read" method, which is: cv2.dnn.readNetFrom Caffe. After the model is fetched from the disk, the advanced method is used to spread the image until the actual object recognition is obtained.

Figure 10 through Figure 15 illustrates when the Intelligent Robot BNU 4.0 is tested or when used for data collection. Including the results of the objects recognition. The data collection distance for object recognition between the robot and the object is carried out at a distance of about 15 to 30 centimeters.



Fig 10. Programming the Arduino Smart Robot BNU 4.0 via a USB connection

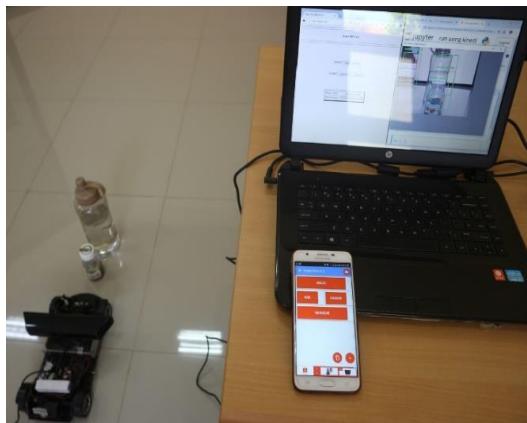


Fig 11. BNU 4.0 Intelligent Robot testing position and data retrieval



Fig 12. BNU 4.0 Intelligent Robot Testing and data retrieval (continued)



Fig 13. Example of BNU 4.0 Robot object detection results using Deep Learning

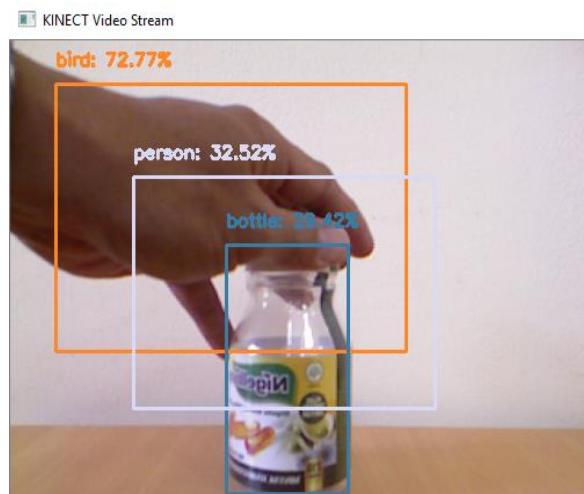


Fig 14. Example result of BNU 4.0 Robot object detection using Deep Learning (continued)

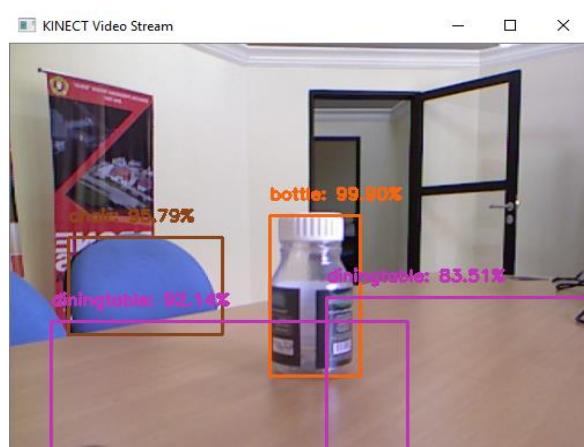


Fig 15. Example result of BNU 4.0 Robot object detection using Deep Learning (continued)

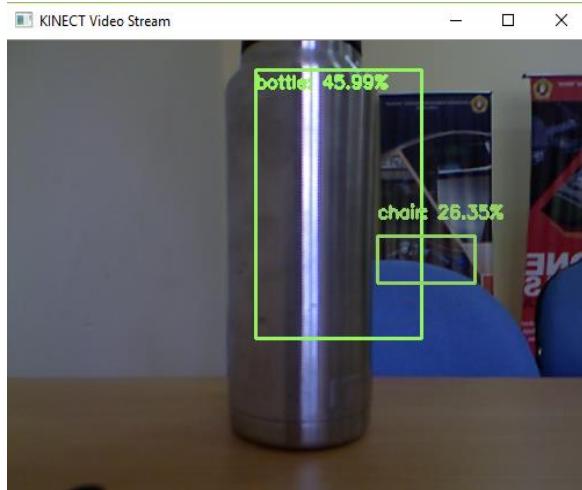


Fig 16. Example result of BNU 4.0 Robot object detection using Deep Learning (continued)

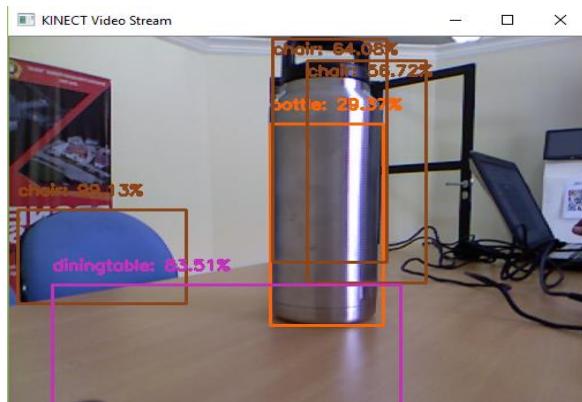


Fig 17. Example result of BNU 4.0 Robot object detection using Deep Learning (continued)

CONCLUSION

It has been explained in this research how to design and create a intelligent surveillance Robot BNU 4.0. This robot control brain is a microcontroller NodeMCU V3. The electronic board based on the ESP8266 chip. With this chip, NodeMCU V3 can connect to the Internet of Things (IoT) cloud. The IoT Cloud used in this research is cloudmqtt (<https://www.cloudmqtt.com>). With the Arduino program embedded in the NodeMCU V3 microcontroller, it can then run web and mobile robot control programs. The mobile robot control program utilizes the Android Apps IoT MQTT Panel. This robot is also equipped with a Kinect Xbox 360 camera. This camera is connected to a computer. With the Deep Learning program using DNN Module from OpenCV and the Python programming language, then the system will recognize the object in front of it.

ACKNOWLEDGMENT

We would like to thank profusely the Ministry of Education and Culture of the Universitas Pembangunan Nasional "Veteran" Jawa Timur for providing research funding under the Letter of Assignment Agreement in the Context of the Implementation of the Independent Research Scheme for the Quality Improvement Learning Program Universitas Pembangunan Nasional "Veteran" Jawa Timur Fiscal Year 2019, Number: SPP / 1 / UN63.8 / LT / IV / 2019.

REFERENCES

- [1] B. Zhang, J. Wu, L. Wang, and Z. Yu, "Accurate dynamic modeling and control parameters design of an industrial hybrid spray-painting robot," *Robot. Comput. Integrat. Manuf.*, vol. 63, p. 101923, 2020.
- [2] S. Pagano, R. Russo, and S. Savino, "A vision guided robotic system for flexible gluing process in the footwear industry," *Robot. Comput. Integrat. Manuf.*, vol. 65, p. 101965, 2020.
- [3] C. Chen, F. Peng, R. Yan, X. Tang, Y. Li, and Z. Fan, "Rapid prediction of posture-dependent FRF of the tool tip in robotic milling," *Robot. Comput. Integrat. Manuf.*, vol. 64, p. 101906, 2020.
- [4] J. Jiang, Z. Huang, Z. Bi, X. Ma, and G. Yu, "State-of-the-Art control strategies for robotic PiH assembly," *Robot. Comput. Integrat. Manuf.*, vol. 65, p. 101894, 2020.
- [5] H. Cao, J. Zhou, P. Jiang, K. K. B. Hon, H. Yi, and C. Dong, "An integrated processing energy modeling and optimization of automated robotic polishing system," *Robot. Comput. Integrat. Manuf.*, vol. 65, p. 101973, 2020.
- [6] K. Lin, Y. Li, J. Sun, D. Zhou, and Q. Zhang, "Multi-sensor fusion for body

sensor network in medical human–robot interaction scenario,” *Inf. Fusion*, vol. 57, pp. 15–26, 2020.

[7] L. Nicholls and Y. Strengers, “Robotic vacuum cleaners save energy? Raising cleanliness conventions and energy demand in Australian households with smart home technologies,” *Energy Res. Soc. Sci.*, vol. 50, pp. 73–81, 2019.

[8] A. Cheong, M. W. S. Lau, E. Foo, J. Hedley, and J. W. Bo, “Development of a Robotic Waiter System,” *IFAC-PapersOnLine*, vol. 49, no. 21, pp. 681–686, 2016.

[9] M. Lanz, R. Pieters, and R. Ghacheloo, “Learning environment for robotics education and industry-academia collaboration,” *Procedia Manuf.*, vol. 31, pp. 79–84, 2019.

[10] P. Kopacek, “Robots in Entertainment, Leisure and Hobby New Tasks for Robot Control,” *IFAC Proc. Vol.*, vol. 33, no. 27, pp. 539–543, 2000.

[11] J. Azeta *et al.*, “An Android Based Mobile Robot for Monitoring and Surveillance,” *Procedia Manuf.*, vol. 35, pp. 1129–1134, 2019.

[12] L. Józwiak, “Advanced mobile and wearable systems,” *Microprocess. Microsyst.*, vol. 50, pp. 202–221, 2017.