



CONFUSION MATRIX USING YOLO V3-TINY ON QUADRUPED ROBOT BASED RASPBERRY PI 3B+

Amperawan Amperawan^{1*}, Destra Andika², Masayu Anisah³, Sabibal Rasyad⁴,
Parni Handayani⁵

¹⁻⁵Department of Electronic Engineering, Politeknik Negeri Sriwijaya, JL.Srijaya Negara
BukitBesar, Palembang, 30139, Indonesia.
amperawan230567@gmail.com

Abstract. The implementation of Confusion Matrix uses Yolo V3-Tiny to detect objects in the form of person cameras and robots for navigation. It is hoped that the quadruped robot in this research can be used during natural disasters so that it can recognize people who are still or have died so that they can be identified. However, in the evacuation process there are often obstacles, such as the absence of information on victims and the location of disaster victims and the difficulty of accessing victims of natural disasters. Based on these problems, research was made regarding quadruped robots by applying a confusion matrix using YOLO V3-Tiny to detect victims of natural disasters. The YOLOv3-tiny model or system can detect person objects according to the dataset that has been studied previously and in this research hardware and software design was carried out by implementing the YOLOv3-Tiny model on a Raspberry PI 3B+ which was placed on a quadruped robot. The system (model) that has been created with YOLOv3-Tiny is capable of detecting person objects. The system (model) created in this research has an Accuracy value of 100%, a Precision value of 1, a Recall value of 1, an F1 – score of 1.

Keywords : Camera, Raspberry PI 3B+, Robot , YOLO

1. Introduction

Based on Law Number 24 of 2007 concerning Disaster Management, a disaster is an event or series of events that threatens and disrupts people's lives and livelihoods caused, either by natural factors and/or non-natural factors or person factors, resulting in person casualties and environmental damage, property loss, and psychological impacts [1-2][8].

The robot is used to detect objects in the form of persons using a special Raspberry camera whose connector is compatible with the Raspberry PI 3B+ board. It is hoped that the robot can detect the presence of people around the disaster in the Raspberry PI using the YOLO algorithm as an object detector. In this research the robot only walks in one room and detects any person. In the future, this robot can be used in natural disasters to determine whether the person detected is still alive or dead [2].

© The Author(s) 2024

N. L. Husni et al. (eds.), *Proceedings of the 7th FIRST 2023 International Conference on Global Innovations (FIRST-ESCSI 2023)*, Advances in Engineering Research 232,
https://doi.org/10.2991/978-94-6463-386-3_56

2. Literatur Reference

In this research, software and software design was carried out to detect robot objects. In its implementation, it requires a reference.

2.1 Robot Quadruped

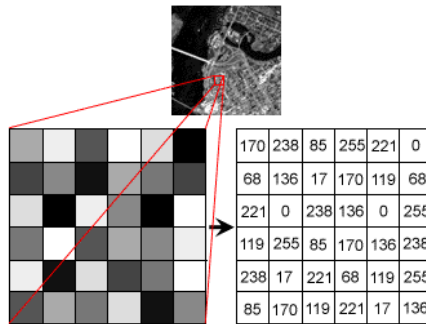
A quadruped robot is a legged robot with four legs. In the design of this quadruped robot leg, each leg has three joints or three DoF (Degree of Freedom) so that the end-effector on the quadruped robot leg can move on the x, y and z axes. Then, to be able to walk or move in place, this quadruped-legged robot has a formulation of leg movement or movement kinematics where the kinematics of this quadruped-legged robot is divided into two, namely forward kinematics and inverse kinematics, apart from that there is also a toe-toe trajectory, gait pattern.) and movement maneuvers so that the robot can move smoothly and precisely [3-5].

2.2 Digital Image

According to Maria, a digital image is an image $f(x,y)$ which has been discretized in spatial coordinates and brightness, $f(x,y)$ means $f(\text{row}, \text{column})$. Digital images are presented by an array or set of two-dimensional arrays where each array represents one color channel. Each element of the array is called a pixel or mop which is taken from the term 'picture element'[3]. A binary image or black and white image is an image that only has 2 possible values for each pixel, namely 0 or 1. The value 0 will appear as black while the value 1 will appear as white. Therefore, this type of image only requires 1-bit to store the value for each pixel. This type of image is often used for masking or image segmentation processes. The image shows the difference between a color image and a grayscale image. To get a binary image, we need a grayscale image that is thresholded based on a specified threshold value. If the pixel value in the grayscale image exceeds or equals the threshold value, then the pixel value is converted to 1. However, if the pixel value is less than the threshold value, then the pixel value is converted to 0.

Grayscale images only have 1 channel so what is displayed is only the intensity value or also known as the degree of gray. This type of image is also called an 8-bit image because each pixel value requires 8-bit storage, namely with a value range of 0 - 255. In theory, there are several ways to convert an RGB color image into a grayscale image, including flattening all the RGB pixels.

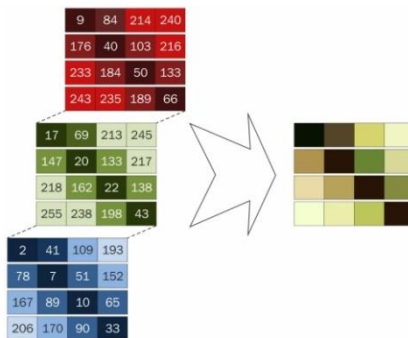
Figure 1 Grayscale images



Color image, the image has channels according to the color space used. The color spaces that are often used are RGB, CMYK, HSV, YCbCr and Lab[6-7].

1. RGB (Red, Green, Blue)

The RGB color space has 3 channels, namely the red channel, green channel and blue channel. Figure 2. is an image of the RGB color space.



Kanal *red* merepresentasikan intesitas warna merah pada citra, kanal *green* merepresentasikan intesitas warna hijau pada citra, sedangkan kanal *blue* merepresentasikan intesitas warna biru pada citra. Rentang nilai pada masing – masing kanal dari 0 – 255 [6-7].

2.3 Raspberry Pi

The Raspberry Pi 3 is the third generation of the Raspberry Pi, replacing the Raspberry Pi 2 Model B in February 2016. The Raspberry Pi3 has an identical build to the previous Raspberry Pi 2 (and the Pi 1 Model B+) and has complete compatibility with the Raspberry Pi 1 and 2. In this newest device, Raspberry adds built-in wireless features and a more powerful processor that was never available in the previous version.

The Raspberry Pi architecture is based around the Broadcom BCM2837 SoC (System-on-a-chip), which has an embedded 1.2GHz 64-bit quad-core ARMv8 processor, VideoCore IV 3D Graphics Core GPU, and 1 Gigabyte of RAM. Data storage is

2.5 PYTHON

Python merupakan bahasa pemrograman yang kuat dan mudah untuk dipelajari yang dibuat oleh Guido van Rossum dan dirilis pada tahun 1991. Python memiliki struktur data efisien yang berlevel tinggi dan pendekatan terhadap pemrograman yang berorientasi objek secara sederhana dan efisien. Python merupakan bahasa yang ideal untuk melakukan system scripting dan pengembangan aplikasi dengan cepat dalam berbagai macam platform. Selain dari itu, Python juga dapat digunakan untuk web development, software development, dan matematika. Penerjemah Python mudah untuk diperluas dengan fungsi dan tipe data baru yang terimplementasi dalam bahasa C. Python juga dapat digunakan sebagai bahasa yang berkelanjutan pada aplikasi yang dapat disesuaikan [6-7] [9].



Figure 5 Python logo

2.6 Camera

The Raspberry Pi foundation launched a 5MP camera which is the OV5647 camera module in 2013. This camera module can be connected to both Raspberry Pi model A and model B using a ribbon connector with a length of 15 cm with 15 lanes. This series can be used with several commands to capture still images or videos. The Raspberry Pi camera module can be seen in Figure 6.



Figure 6. Raspberry Pi Camera Rev 1.3

From the first launch the Raspberry Pi has had a connector for attaching a camera to the GPU (VideoCore 4 Graphics Processing Unit on the Raspberry Pi).

This connection uses the CSI-2 protocol and is the standard used in some cell phones. This connection is capable of sending 1 image measuring 1080p (1920x1080x10bpp) at 30 frames per second. Or lower resolution at higher frame rates, because the ability to stream high-speed video data through the GPU without interaction with the ARM processor makes the camera much more efficient than a USB-mounted webcam[6-7][10].

2.7 SERVO DYNAMIXEL AX-12

Basically, servo motors are components that are often used in the field of robotics, usually servo motors become robot joints or joints in robot legs and arms, making this component very important. It is known that there is a type of dynamixel AX-12A servo motor which can be said to be a servo motor that has higher intelligence than servo motors in general. Dynamixel is a modular intelligent robot actuator that combines a gear reducer, precision DC motor and control circuitry, all in one package. Despite their small size, dynamixel serial servos can produce high torque and are made from high-quality materials to provide the necessary strength and durability to withstand large external forces.

This dynamixel serial servo also has the ability to detect and act on internal conditions such as changes in temperature or supply voltage. Dynamixel actuators have many advantages compared to similar products. Position and speed can also be controlled serially with a resolution of 1024 steps, the AX-12A dynamixel servo can be seen in Figure 7[4][9][11].

Figure 7 Servo AX-12A

2.8 Confusion Matrix

Confusion matrix is also called error matrix, which means it provides information comparing the classification results carried out by the system (model) with the actual classification results. The confusion matrix is in the form of a matrix table that describes the performance of the classification model on a series of test data where the actual values are known, which can be seen in Table 1. There are 4 terms that represent the results of the classification process in the confusion matrix, namely True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) [3].

The following is an explanation of the 4 terms in the confusion matrix table:

- True Positive (TP), is positive data that is predicted correctly by the system (model). For example, there is a person object and the system (model) detects that the object is a person object.
- True Negative (TN), is negative data that is predicted correctly by the system (model). For example, there is an object and the system (model) does not detect that the object is an object.
- False Positive (FP), is negative data that is predicted to be positive by the system (model). For example, there is an object and the system (model) detects that the object is a person object.
- False Negative (FN), is positive data that is predicted to be negative by the system (model). For example, there is an object and the system (model) detects that the object is an object.

		<i>Actual Values</i>	
		1 (Positive)	0 (Negative)
<i>P</i>	1	TP	FP
	0	FN	TN

		FN	TN
--	--	----	----

Table 1. Confusion Matrix

Berikut ini merupakan tujuan dan manfaat dari *confusion matrix* :

- a. Menunjukkan bagaimana sistem (model) yang dibuat membuat suatu prediksi.
- b. Memberi informasi tentang kesalahan yang dibuat dan jenis kesalahan yang dibuat oleh sistem (model).
- c. Setiap kolom dari *confusion matrix* merepresentasikan *instance* dari kelas prediksi.
- d. Setiap baris dari *confusion matrix* mewakili *instance* dari kelas aktual.

Information :

- TP = Number of person objects detected
- FN = Number of person objects detected as objects.
- FP = Number of objects detected as person
- TN = Number of objects detected

In the equation below, TP, FP, FN, TN are the results of the performance metric calculation.

1. *Accuracy*

For the accuracy value formula with the equation:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \times 100\% \quad (1)$$

From the calculation formula above, it can be seen that the system (model) is able to detect objects in the form of person s with an accuracy value of 100%.

2. *Precision*

For the *Precision* value formula with the equation:

$$Precision = TP / (TP + FP) \quad (2)$$

From the formula you can produce a system (model) that has a precision value of 1.

3. *Recall*

For the *recall* value formula with the equation:

$$Recall = TP / (TP + FN) \quad (3)$$

4. *F1-score*

For the *F1-score* value formula with the equation:

$$F1-Score =$$

$$2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (4)$$

3. Experiment

The design of the system made hardware such as the block diagram below:

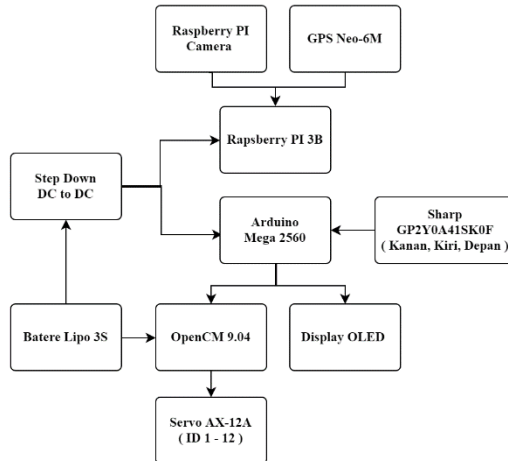


Figure 8 System Block Diagram

This research can now be implemented in the form of a Quadruped Robot. A Quadruped Robot is installed with a camera and motor servo to detect people and navigate to avoid surrounding objects other than people.



Figure 9 Quadruped Robot implementation

This test is carried out to find out whether the training results can detect objects in the form of person or can also detect not person objects. This test needs to be carried out to ensure that the person object will be detected. This test is carried out by inputting data in the form of a video and then the video is detected. This is done 15 times with different videos. Video downloaded from www.istockphoto.com

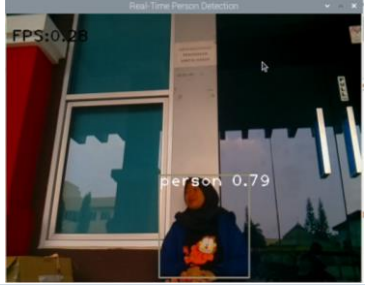
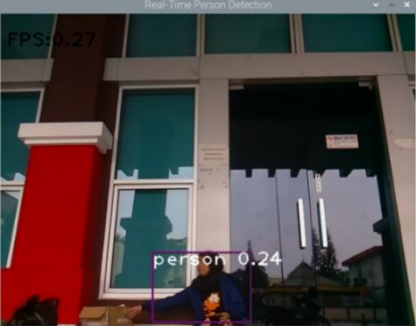
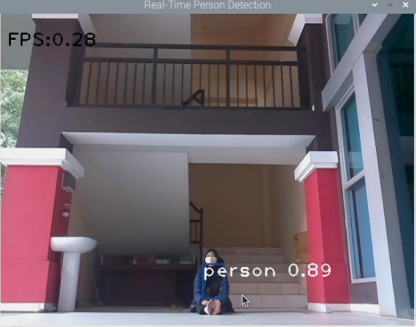
Table 2. Object Detection Testing

Test Data	Number of person in the video	Number of person detected	The number of person is undetectable
Video 1	7	7	0
Video 2	5	4	1
Video 3	9	5	4
Video 4	5	5	0
Video 5	5	5	0
Video 6	8	8	0
Video 7	6	3	3
Video 8	5	5	0
Video 9	4	3	1
Video 10	7	7	0
Video 11	13	13	0
Video 12	2	2	0
Video 13	5	5	0
Video 14	7	6	1
Video 15	9	9	0
Jumlah	97	87	10

This test was carried out to find out whether the training results could detect objects in the form of persons at a distance of 2 meters to 5 meters. Table 1 for Person object testing.

Table 3. Object Distance Testing

Object Distance and detection	Object (Person) Detection
2 Meter/ Detection	

3 Meter/ Detection	
4 Meter/ Detection	
5 Meter/ Detection	

4 Result And Discussion

From the test results, performance metrics can be calculated from table 1:

1. Accuracy

To get the accuracy value, a calculation is carried out using equation that the results are:

$$\begin{aligned}
 \text{Accuracy} &= \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \\
 &= \frac{87+0}{87+0+0+0} \times 100\% \\
 &= 100\%
 \end{aligned}$$

From the calculations above, it can be seen that the system (model) is able to detect objects in the form of person s with an accuracy value of 100%.

2. Precision

To get the precision value, a calculation is carried out using equation that the results are obtained:

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP+FP} \\ &= \frac{87}{87+0} \\ &= 1 \end{aligned}$$

From the calculation above, it can be seen that the system (model) has a precision value of 1.

3. Recall

To get the *Recall* value, a calculation is carried out using equation that the results are obtained:

$$\begin{aligned} \text{Recall} &= \frac{TP}{TP+FN} \\ &= \frac{87}{87+0} \\ &= 1 \end{aligned}$$

From the calculation above, it can be seen that the system (model) has a *recall* value of 1.

4. F1-score

To get the *F1-score* value, a calculation is carried out using equation that the results are obtained:

$$\begin{aligned} \text{F1-Score} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \\ &= 2 \times \frac{1 \times 1}{1+1} \\ &= 1 \end{aligned}$$

From the calculation above, it can be seen that the system (model) has a *F1 - score* value of 1.

5. Conclusions

From this research, the results of the tests that have been carried out prove that the system (model) that has been created with YOLOv3-tiny is capable of detecting person. The system (model) created in this research has an Accuracy value of 100%, a Precision value of 1, a Recall value of 1, an F1 – score of 1.

Reference

1. P. R. Indonesia, *Undang-Undang Republik Indonesia Nomor 24 Tahun 2007 Tentang Penanggulangan Bencana*, vol. 9, no. 2. 2007.
2. Menteri Dalam Negeri Republik Indonesia, *Menteri Dalam Negeri Republik Indonesia Peraturan Menteri Dalam Negeri Nomor 33 Tahun 2006*, vol. 1999, no. December. 2006.
3. D. R. Hartawan, T. W. Purboyo, and C. Setianingsih, “Disaster victims detection system using convolutional neural network (CNN) method,” *Proc. - 2019 IEEE Int. Conf. Ind. 4.0, Artif. Intell. Commun. Technol. IAICT 2019*, pp. 105–111, 2019, doi: 10.1109/ICIAICT.2019.8784782.
4. W. T. Wibowo, “Implementasi Inverse Kinematics Pada Pergerakan Robot Quadraped,” Surabaya, 2017.
5. R. Syam, “Kinematika dan Dinamika Robot Lengan,” p. 76, 2015.
6. P. Hidayatullah, *Pengolahan Citra Digital Teori dan Aplikasi Nyata*. Bandung: Informatika Bandung, 2017.
7. F. D. Marleny, *Pengolahan Citra Digital Menggunakan Python*. Banjarmasin, 2021.
8. Badan Nasional Penanggulangan Bencana (BNPB), *Peraturan Kepala Badan Nasional Penanggulangan Bencana Nomor 13 Tahun 2010 Tentang Pedoman Pencarian, Pertolongan dan Evakuasi*. 2010.
9. I. Sugianda and Thamrin, “PERANCANGAN SISTEM DETEKSI OBJEK PADA ROBOT KRSBI BERBASIS MINI PC RASPBERRY PI 3,” *J. Teknol. Inf. dan Pendidik.*, 2019.
10. X. Shao, Y. Yang, and W. Wang, “Obstacle Crossing with Stereo Vision for a Quadraped Robot,” *Proc. 2012 IEEE*, 2012.
11. Z. Nasution, A. F. Ilham Suparman, G. A. Prasetyo, A. H. Alasiry, E. H. Binugroho, and A. Darmawan, “Body Balancing Control for EILERO Quadraped Robot while Walking on Slope,” *IES 2019 - Int. Electron. Symp. Role Techno-Intelligence Creat. an Open Energy Syst. Towar. Energy Democr. Proc.*, pp. 364–369, 2019, doi: 10.1109/ELECSYM.2019.8901598

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

