DESIGN AND IMPLEMENTATION OF ROBOMAGELLAN SHORT-NAVIGATION USING DEEP LEARNING AND FUZZY LOGIC

Dena Meidina[1], Taufiq Nuzwir Nizar[2], Rodi Hartono[3]

[1,2]Informatic Engineering – Indoensia Computer University
Jl. Dipatiukur 112-116, Bandung 40132, Indonesia
E-mail : dena05meidina@email.unikom.ac.id[1], taufiq.nizar@email.unikom.ac.id [2], rodi.hartono@email.unikom.ac.id [3]

ABSTRAK

This study aims to develop a system of avoiding humans by using deep learning in autonomous car robot prototypes. The system detects human presence and calculates the distance of circumstances around. The system for humans will use deep learning with the YOLOv3 algorithm. The system for calculating the distance of circumstances around will use ultrasonic sensors. The system applied has an average avoidance accuracy of 85.71% with an ideal distance to avoid more than 2 meters. The use of YOLOv3 is good enough to be applied to an autonomous car in object detection.

Keyword : robot, prototype, deep learning, YOLOv3, autonomous car.

1. PENDAHULUAN

Over the past fifty years, researchers have continued to look for keys to minimizing human input in driving. Jansson stated that 93% of car accidents were caused by human error and research conducted by Lebanese Red Cross revealed that car accidents in 2014 resulted in 14,516 fatalities. This shocking statistic is caused by the ever-increasing traffic density in slow-developing infrastructure as said by Zlocki et al. From these results it results in more complex and more difficult driving which results in an increase in the possibility of human error and increases the accident rate [1].

The need for an Autonomous car is needed to reduce or even eliminate driving accidents from human error. Navigation of robots in a dynamic environment is still a challenge for real-world implementation. The robot must be able to obtain its objectives by navigating safely among people or vehicles that are moving, facing implicit uncertainty from the surrounding environment and the perceptions of the system's boundaries [2].

One method of detecting humans is using HOG and SVM [3] [4], autonomous car remote navigation research [5], cone detection [6] and self driving car research [7]. In order to develop this research, it will be specific to avoid humans. The system of detecting humans will use deep learning with the YOLOv3 algorithm and calculate the distance around using a proximity sensor that will be used to avoid humans.

2. LITERATURE REVIEW

2.1 Ackerman Steering System

Consideration of the Ackerman steering system used in this car is very common in the real world but is unusual in the robot world. Ackerman's limited steering maneuverability has superior directional direction and steering geometry in lateral stability at high speeds. The steering system like this car is one of the simplest movement systems where the motor drive and turning turn are separate. Under one condition where the turning mechanism must be precisely controlled a small position error in the turning mechanism will cause a large odometric error. The weakness of the Ackerman system is that it cannot rotate the vertical axis and turning radius which is limited by the turning angle of the front wheel and the distance between the rear wheel and the front wheel [8]. Following is the ackerman steering system can be seen in Figure 1.

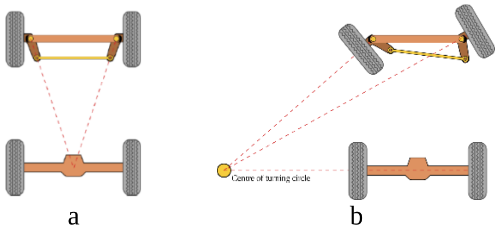


Figure 1. Ackerman Steering System, Steering System (a), Steering Wheel (b)

2.2 Ultrasonic Sensor

Ultrasonic sensors consist of two transmitter circuits, namely the transmitter circuit and the receiver circuit. Ultrasonic waves are acoustic waves that have frequencies from 20 kHz to 40 kHz. The working principle of the ultrasonic sensor is that when the transmitter emits ultrasonic waves at the same time the sensor will produce an output of an upward transition indicating the sensor begins to count time, after the receiver receives the reflection generated by an object, the time will be stopped by producing a downward transition output. If time is t and the speed of sound is 340 m / s, the distance between the sensor and the object is calculated.

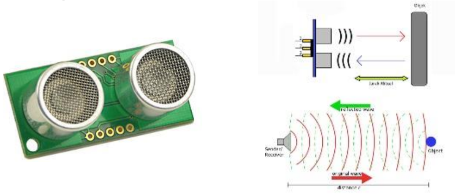
$$s = \frac{t * 340\ {}^{m}/_{s}}{2}$$

Where:

s = the distance between the sensor and the object (meters)

t = ultrasonic wave time from transmitter to receiver (s)

Following is an example of an ultrasonic sensor and the working principle of an ultrasonic sensor can be seen in Figure 2.



a b

Figure 2. Ultrasonic Sensors, Examples of Ultrasonic Sensors (a), Working Principles of Ultrasonic Sensors (b)

2.3 Deep Learning

Deep learning is one of the machine learning algorithms by using several layers to progressively extract high-level features from raw inputs. Deep learning overcomes the problem of representation by introducing representations that are expressed in the form of other, simpler representations. Examples of deep learning models are feedforward deep network or multilayer perceptron (MLP). An MLP is simply a mathematical function that maps several sets of input values to output values [9]. Here is how the deep learning system can represent the concept of a person's image by combining simpler concepts can be seen in Figure 3.

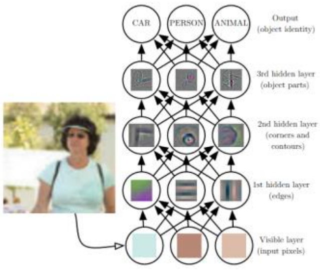


Figure 3. Illustration of Deep Learning Model

From Figure 3 the computer is difficult to understand the meaning of the raw input data. The image is represented as a set of pixel values. Mapping functions from a set of pixels to object identity is very complicated. Evaluation of this mapping seems insurmountable if it is handled directly. Deep learning solves this difficulty by breaking down complex mappings into a series of simpler nested mappings, each explained by a different layer of the model. Input is input in the visible layer, because it can be observed. Then the hidden layer extracts abstract features from the image. Named hidden layer because the value does not give data so the model must determine the choice of concepts that are useful to explain the relation of the observed data. The first hidden layer can easily identify the margins by comparing the brightness of pixel by pixel. The second hidden layer easily looks for angles and contours which are obtained from the identified outlines. The third hidden layer can detect several parts of an object, by finding specific objects from angles and contours. So finally the description of the object it contains can be used to recognize objects in the picture.

2.4 YOLOv3

YOLO (You Only Look Once) is a deep learning object detection algorithm that is targeted for real-time processing [10]. The working system of YOLOv3 is by repeating the system classifiers or localizers to do the detection. By applying the model to an image in various locations and various scales. The area with the highest value of an image will be considered as a detected object.

By using a completely different approach. YOLO applies a single neural network to an image. The network is divided into several regions (regions) and predicts bounding boxes and looks for probabilities for each region (regions). With bounding boxes that are weighted from predicted probabilities.

3. METHOD AND DISCUSSION

3.1 System planning

The design of the robomagellan melee navigation system will use deep learning to detect humans and the algorithm used is YOLOv3 (You Only Look Once) and YOLOv3 provides training data that already exists to detect humans, after knowing humans are detected, then calculates the robot's angle to humans, then the three-way ultrasonic sensor will be used to determine the distance of the object from the left, center and right of the robot so that the distance and angle of the detected human being input fuzzy logic Tsukamoto method. The defuzzification results from fuzzy logic will be used to determine the angle of servo steering and the speed of the robot avoiding humans. Following is a System Diagram Block can be seen in Figure 4.

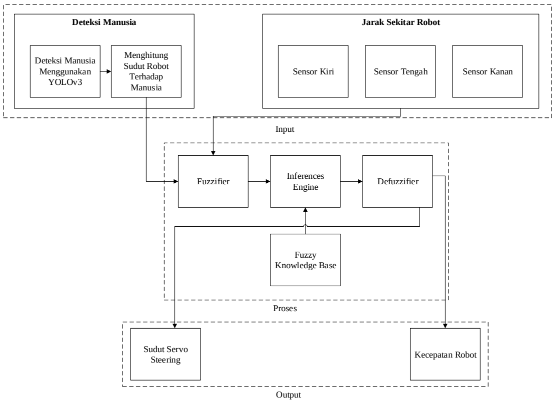
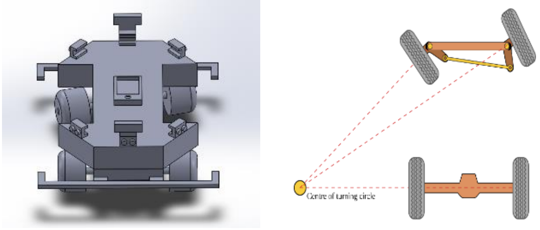


Figure 4. System Diagram Block

3.2 Navigation Steering System Jelajah V-18

Navigation steering system used Cruising V-18 is using the Ackerman steering system which is the simplest drive system where the driving motor is separate from the turning system. There is a condition where the turning mechanism must be controlled precisely, small turning position errors in the rotation mechanism can cause large odometry errors. This system uses input media from the result of defuzification to the servo motor as its steering. For more detailed design that will be built will be shown in Figure 5.



a b

Figure 5. Steering System Steering-V18, Steering Design Jelajah V-18(a) , Arckerman Steering System (b)

3.3 Human Detection System

If humans are detected it will continue to calculate angles, if humans are detected by more than one person, data will be detected with the largest bounding box. Then to get the angle of the object from the data obtained the formula:

$$s = (xo / t) * F$$

where :

s = the robot's angle to humans is detected

xo = the difference from the center point of the camera to the midpoint of humans detected

t = the midpoint of the camera resolution

F = The camera's Field of View is divided in two

Here is a picture of looking for an angle can be seen in Figure 6.

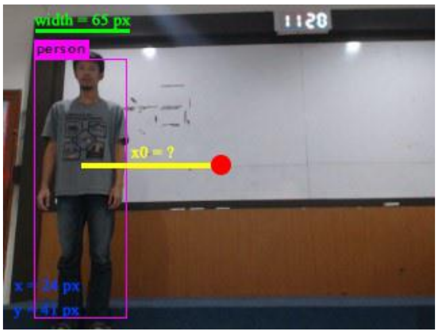


Figure 6. Looking for Angles

From Figure 6 which has a resolution of 320x240 px obtained data x = 24, y = 41 and width = 65, then to find the angle, must find the length from the midpoint of the camera to the midpoint of the detected human (xo).

$$\begin{aligned} xo &= t - (x + width / 2) \\ &= 160 - (24 + 65/2) \\ &= 160 - 57.5 \\ &= 102.5 \end{aligned}$$

Next determine the angle to be sought.

$$\begin{aligned} s &= (xo / t) * F \\ &= (102.5 / 160) * 39 \\ &= 24.984 \end{aligned}$$

From the calculation results, the robot's angle to humans was detected at 24,984 ° on the left.

3.5 Main Algorithm

The main algorithm is an overall robot algorithm to solve the avoidance that will be carried out by robomagellan in avoiding humans. In general, the main algorithm flow will be shown in Figure 7.

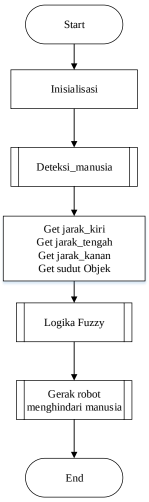


Figure 7. Main Algorithm Flowchart

3.6 Human Detection Algorithm

The human detection algorithm will use the YOLOv3 algorithm if humans are detected then the midpoint of the position of the camera resolution will be taken and then processed to produce angles as input data to proceed to the fuzzy logic process. The flow of human detection algorithm can be seen in Figure 8.

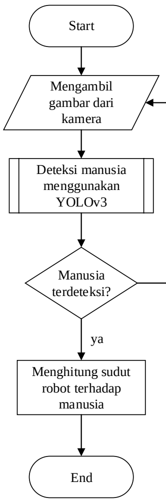


Figure 8. Human Detection Algorithm Flowchart

3.7 Robot Motion Algorithm Avoiding Humans

The robot motion algorithm is used when the robot is switching from remote navigation to close navigation when humans are detected in front of the robot. The robot motion algorithm can be seen in Figure 9.

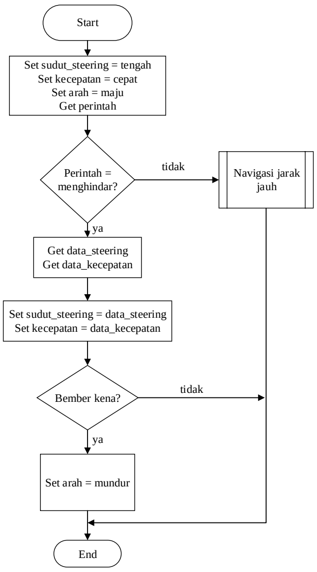


Figure 9. Robot Motion Algorithm Flowchart

4. RESULT

4.1 Human Detection Testing

In the DarkNet framework model configurations and weights are available that can be used to detect humans, so this research will compare two of the several configurations that can be applied with hardware specifications that exist in robots, testing is done with distance parameters, fps, accuracy of detecting humans, and success in detecting humans. Testing the configuration can be seen as follows:

1. Yolov3 configuration

The standard model and weight of the DarkNet framework uses the yolov3 configuration as the initial configuration. The following results of testing the model configuration and yolov3 weight can be seen in Table 1.

Table 1. Model Configuration Test Results and Yolov Weigth3

Distance (m)	Success in detecting humans (yes / no)	Accuracy in detecting humans (%)
1	yes	98
2	yes	100
3	yes	100
4	yes	100
5	yes	100
6	yes	100
7	yes	100

By using the model configuration and yolov3 weight average fps is 3.5 fps.

2. Yolov3-tiny configuration

The yolov3-tiny configuration is a very light model and weight that increases fps but reduces accuracy. The following results of testing the model configuration and yolov3-tiny weight can be seen in Table 2.

Table 2. Yolov3-tiny Model and Weight Test Results

Distance (m)	Success in detecting humans (yes / no)	Accuracy in detecting humans (%)
1	no	0
2	yes	87
3	yes	95
4	yes	99
5	yes	98
6	yes	97
7	yes	85

By using the model configuration and yolov3-tiny weight the average fps is 25 fps.

The conclusion from the comparison of the two model configurations and weight adjusts to the needs of the system, therefore the system will use the model and weight configuration of yolov3-tiny because in close-range navigation systems, real-time data needs high fps to avoid humans.

Table 3. Angular Accuracy Testing Results

No	Real angle (°)	The angle of calculation results (°)	Error (°)
1	0	1.82	1.82
2	10	8.165	1.835
3	20	18.281	1.719
4	30	34.368	4.368
5	40	0	0
		Average Error	2.4355

Based on Table 3 obtained at an angle of 40 humans are not visible on the camera so the camera can only detect humans a maximum of 30 ° to the left and 30 ° to the right. With an average error of 2.4355 for the system can still avoid humans properly.

4.2 System Testing Results on Robomagellan Jelajah V-18

Testing includes success from avoiding humans. Testing is done in an open area, testing is done by calculating the distance between robots and humans with human conditions do not move and calculate the success of avoiding humans from a distance that has been determined, each distance tested is done each of five experiments. Criteria in testing are said to be successful when the robot manages to avoid passing humans without crashing. The following results of testing the short-range navigation system can be seen in Table 4.

Table 4. System Testing Results

Distance (m)	Success (%)
1	20
2	80
3	100
4	100
5	100
6	100
7	100

Based on the test results in Table 4 that by using a model configuration and yolov3-tiny weight obtained the percentage of success is 85.71% and the ideal distance to avoid humans is at a distance of two or more so that it does not crash.

5. CONCLUSION

5.1 Conclusion

From research, implementation and testing, the writer can conclude a short-range navigation system, namely:

1. The system can detect humans well starting from a distance of two meters with the condition of some limbs seen in the picture.
2. The system can safely avoid humans without crashing from a distance of 3 meters.
3. The system is able to adjust the steering speed and angle well when humans are already selected.
4. The maximum camera can detect humans is only 30 ° from middle to left and 30 ° from middle to right.

5.2 Suggestions

To develop this short-distance navigation system so that the level of accuracy avoids humans, the writer's suggestion is:

1. Replacing the specifications of a mini PC with a mini PC that already has a GPU (Graphics Processing Unit) to increase the performance of the detection system.
2. Adding other objects that can be detected as obstacles, can be done with training data first.
3. Replacing the ultrasonic sensor with the LIDAR (Light Detection and Ranging) senor to improve accuracy, reading speed and maximum distance.
4. Replace the camera with a wider Field of View so that the angle you can get can be even wider.

DAFTAR PUSTAKA

[1] Z. Yassine, D. Mohamad, M. Ossama. 2018. “GPS Tracking System For Autonomous Vehicles”. Alexandria Enginerring Journal. 57(4): 3127-3137.

[2] B. Hadjira, B. Hamid, B. Mohamed. 2016. “Implementation Of a Mobile Robot Platform Navigating In Dynamic Environment”. ICMME. 95.

[3] N.N. Taufiq, A. Nurfitri, S. P. Ary, 2014. “Multi-Object Tracking and Detection System based on Feature Detection of the Intelligent Transportation System”, IEEE International Conference on System Engineering and Technology. ICSET.

[4] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, 2005, vol. 1, pp. 886–893 vol. 1.

[5] R. Imam, A.J. Didit, “Rancang Bangun Sistem Kendali Gerak pada *Robomagellan* (Jelajah – V18)” , Universitas Komputer Indonesia, Bandung.

[6] P.W. Feki, A.J. Didit, “Rancang Bangun Sistem Pendeteksi *Cone* pada *Robomagellan* (Jelajah – V18)” , Universitas Komputer Indonesia, Bandung.

[7] Sujeetha, B. Chitrak, P. Gareja, P. Siddhant, 2019. “THE YAN: [Self Driving Car Uisng Deep Learning]”, International Journal of Applied Engineering Research.Volume 4, Number 9 : pp. 2156-2161.

[8] G. Kristof, 2004, “Autonomous Mobile Robot Mechanical Design”, Vrije Universiteit Burssell, Academiejaar.

[9] G. Ian, B. Yoshua, C. Aaron, 2016, “Deep Learning (Adaptive Computation and Machine Learning Series)”, MIT Press.

[10] R. Joseph, F. Ali, 2018, “YOLOv3: An Incremental Improvement”, arXiv: 1804.02767.