Jurnal Rekayasa Elektrika Vol. 17, No. 2, Juni 2021, hal. 115-122
ISSN. 1412-4785; e-ISSN. 2252-620X, Terakreditasi RISTEKDIKTI No. 30/E/KPT/2018
DOI: 10.17529/jre.v17i2.20805

Improved Performance of Trash Detection and Human Target Detection Systems Using Robot Operating System (ROS)

Kisron[1], Bima Sena Bayu Dewantara[1], and Hary Oktavianto[2]
[1]Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya
[2]Department of Electrical Engineering, Politeknik Elektronika Negeri Surabaya
Jl. Raya ITS, Keputih, Kec. Sukolilo, Surabaya, Jawa Timur 60111
e-mail: kisron@pasca.student.pens.ac.id

***Abstract*—In a visual-based real detection system using computer vision, the most important thing that must be considered is the computation time. In general, a detection system has a heavy algorithm that puts a strain on the performance of a computer system, especially if the computer has to handle two or more different detection processes. This paper presents an effort to improve the performance of the trash detection system and the target partner detection system of a trash bin robot with social interaction capabilities. The trash detection system uses a combination of the Haar Cascade algorithm, Histogram of Oriented Gradient (HOG) and Gray-Level Co-ocurrence Matrix (GLCM). Meanwhile, the target partner detection system uses a combination of Depth and Histogram of Oriented Gradient (HOG) algorithms. Robotic Operating System (ROS) is used to make each system in separate modules which aim to utilize all available computer system resources while reducing computation time. As a result, the performance obtained by using the ROS platform is a trash detection system capable of running at a speed of 7.003 fps. Meanwhile, the human target detection system is capable of running at a speed of 8,515 fps. In line with the increase in fps, the accuracy also increases to 77%, precision increases to 87,80%, recall increases to 82,75%, and F1-score increases to 85,20% in trash detection, and the human target detection system has also improved accuracy to 81%, %, precision increases to 91,46%, recall increases to 86,20%, and F1-score increases to 88,42%.**

Keywords: ***ATRACBOT, performance improvement, robot operating system, trash detection, target detection***

***Abstrak*—Pada sebuah sistem deteksi riil berbasis visual menggunakan visi komputer, hal terpenting yang harus diperhatikan adalah waktu komputasi. Pada umumnya, sebuah sistem deteksi memiliki algoritma yang berat sehingga membebani kinerja sistem komputer, terlebih jika komputer harus menangani dua atau lebih proses deteksi berbeda. Makalah ini menyajikan upaya peningkatan performa sistem deteksi sampah dan sistem deteksi target partner sebuah robot tempat sampah dengan kemampuan interaksi sosial. Sistem deteksi sampah menggunakan kombinasi algoritma Haar-Cascade, *Histogram of Oriented Gradient* (HOG) dan *Grey-Level Coocurrence Matrix* (GLCM). Sedangkan sistem deteksi target partner menggunakan kombinasi algoritma *Depth dan Histogram of Oriented Gradient* (HOG). *Robotic Operating System* (ROS) digunakan untuk membuat setiap sistem dalam modul-modul terpisah yang bertujuan untuk memanfaatkan seluruh sumber daya sistem komputer yang ada sekaligus mengurangi waktu komputasi. Sebagai hasilnya, performa yang didapatkan dengan menggunakan platform ROS ini adalah sistem deteksi sampah mampu berjalan dengan kecepatan 7,003 fps. Sedangkan sistem deteksi target manusia mampu berjalan dengan kecepatan 8.515 fps. Sejalan dengan peningkatan fps tersebut, akurasi juga naik menjadi 77%, presisi naik menjadi 87,80%, *recall* naik menjadi 82,75%, dan F1-*score* naik menjadi 85,20% pada deteksi sampah, dan sistem deteksi target manusia juga memperbaiki akurasi menjadi 81%, presisi naik menjadi 91,46%, *recall* naik menjadi 86,20%, dan F1-*Score* naik menjadi 88,42%.**

Kata kunci: ***ATRACBOT, peningkatan performa, robot operating system, deteksi sampah, deteksi target***

I. Introduction

ATRACBOT is a smart trash caring robot that was built to provide education on the importance of disposing of trash in its place to early childhood. This education really needs to be done because the data in the Indifference Behavior Index for the Environment According to Provinces in Java-Bali Island, Central Java and East Java Provinces have an index of 0.75 in the dimension of trash management. [1][2]. The robot is developed by starting from planning, sub-system creation, integration and evaluation. The robot is designed to have several capabilities. Several of them are detecting and classifying trash objects [3], detecting human target around the robot

Received 20 April 2021; Revised 13 June 2021 ; Accepted 18 June 2021

[4], communicating with the officer [5][6], and navigating around the social environments [7].

Those abilities have been carried out by previous researchers, namely the detection and classification of trash objects by Salimi *et.al.* [3] with the detection results performance up to 73.49% with 3.221 fps. Then, the human target and persuasive social interaction was carried out by Dewantara *et.al.* [4] which achieves 71% of detection results with 4.26 fps. For robot monitoring and communication with the officer, Kisron *et.al.* [5][6] has developed a IoT-based system which achieves 100% of accuracy, 100% of precision and 100% of recall results for the robot's decision-making system. The Telegram bot-based application is able to send and receive information between the robot and the officer. The navigation system [7] is developed in the form of simulation, and the robot has not been implemented directly.

Salimi *et.al.* [3] developed a visual-based system that can detect the presence of trash by using Haar-Cascade. Then, a combination of Gray-Level Co-Occurrence Matrix (GLCM) and Histogram of Oriented Gradient (HOG) is used as the features to be fed to the Support Vector Machine (SVM) to distinguish the type of trash into organic and non-organic. The system was developed under standard C++ programming on Windows OS platform without any optimization techniques. The main drawback of the system is very low computing speed. This is the main cause of the low processing speed of 3.221 fps.

On the other hand, the second ability of the robot is to detect the presence of human around the robot. This feature will work when the robot has managed to find the trash and will ask someone for help to pick up and put the trash into the bin carried by the robot. Dewantara *et.al.* [4] proposed Depth-HOG as the shape feature of a human upper body that is classified using SVM to distinguish human or not human. The system was also developed under standard C++ programming on Windows OS platform without any optimization techniques. The main drawback of the system is very low computing speed. This is also the main cause of the low processing speed of 4.26 fps.

Another problem that arises is when multiple abilities are supposed to work together to achieve a goal cannot be combined easily, then the robot system as a whole will not work optimally. Therefore, we propose using ROS for overall system integration in order to maximize existing resources to build a reliable system and converting the operating system to Linux in order to further reduce the computation time.

This paper will explain the performance evaluation of the two sub-systems that have been made, namely the trash detection and classification system and the human target detection system. Evaluation of the system is carried out by comparing the platforms used. In addition to performance comparisons, system integration is also carried out so that the two sub-systems can run simultaneously.

II. Literature Review

To get maximum research results, the researcher made comparisons of several methods. This is done so that the method used is in accordance with the characteristics of the system because each system has different characteristics. Asim Roy *et al.* [8] have compared several methods and evaluated machine learning algorithms based on performance measures (e.g., Accuracy, Area Under the Curve (AUC), and F-score). Such a method can be used to compare standard Machine Learning platforms such as SAS, IBM SPSS, and Microsoft Azure ML.

They compare platforms based on predictive performance on classification problems because most of the problems in machine learning are those of the platform type. Common questions posed include the following: Is there any platform that outperforms the others on certain performance measures? For each platform, they use a set of six classification algorithms from the following six algorithm families - support vector machine, multilayer perceptron, random forest, random trees / gradient boost trees, naive Bayes/Bayesian network, and logistic regression. To test the platform, they used several datasets from the University of California library, Irvine (UCI), Kaggle Competition Library, and high-dimensional gene expression problems. They also perform parameter tuning of the algorithm. The results obtained by each method have different performance when applied to different cases.

Ahmad Ashari *et.al.* [9] proposed a new method in finding design alternatives, namely by using the classification method. The classification methods they use are Naïve Bayes, Decision Tree, and k-Nearest Neighbor. Their experiments show that the Decision Tree has the fastest classification time followed by Naïve Bayes and k-Nearest Neighbor. The difference between the Decision Tree and Naïve Bayes classification times as well as between the Naïve Bayes and k-NN is about the order of magnitude. Based on Percision, Recall, F-measure, Accuracy, and AUC, Naïve Bayes' performance is the best.

Aldi Kika and Silvana Grecca [10] explored the performance of a Java image processing application designed with a multithreading approach. To test how multithreading affects program performance, they tested several image processing algorithms implemented in Java using a sequential one thread approach and multithreading on single and multi-core CPUs. Experiments are based not only on different platforms and algorithms which differ from each other by their level of complexity, but also on changes in image size and thread count when the multithreading approach is applied. Performance is improved on single core and multiple CPU cores in different ways due to image size, algorithm complexity and platform.

The results showed that the multithreading approach improved the performance of the algorithm processing time on both single-core and multi-core CPU platforms, but this increase was different. In a single core, the best results are given by a combination of small image sizes

and less complex algorithms, whereas on multicore CPUs the combination of small image sizes and more complex algorithms improves performance. Multithreading programming can improve performance on multi-core CPUs when complex image processing algorithms are applied.

III. Metodology and System Design

A. Trash Detection and Classification

ATRACBOT has the ability to detect and differentiate trash into organic or non-organic types. The initial research by Salimi *et al.* [3] developed a detection system using a single core CPU programming and gets the detection results with an accuracy of 73.49% and fps of 3,221 fps. Fig. 1 shows the system design of the trash detection system on ATRACBOT that has been developed in [3].

Based on the block diagram in Fig. 1, the system's input is an image frame with a scene contains trash object obtained from the RGB camera. Preprocessing is done by reducing the image size so that it is faster to process. Object detection is built using the Haar-cascade method [11]. For the training purpose, we used a set of positive images such as bottles, cans, paper bundles, drinking boxes, leaves, and plastic wraps. While negative images are taken from objects other than the positive images.

The Haar-cascade is used to determine the features by processing the image by using a set of black-white squares that has a certain pixel size. The results are then fed to an integral image process. And finally, the result of integral image is classified as trash or not using a combination of some Adaptive Boosting in the form of Cascaded Classifier.

After the trash is detected, then, the type of trash should be classified. We use a texture and shape-based features by utilizing the Gray-Level Co-Occurrence Matrix (GLCM) and Histogram of Oriented Gradient (HOG), respectively. The GLCM will calculate the variation of the neighboring values of the image and then create a matrix. From the matrix that has been formed, the value of each texture feature can be calculated. There are six values that will be the GLCM features, namely Contrast, Inverse Different Moment (IDM), Energy, Entropy, Homogeneity, and Mean Square Error (MSE). Each value is formulated as follows.

$$Contrast = \sum_{n=1}^{L} n^2 \left\{ \sum_{|i-j|=n} P(i,j) \right\} \quad (1)$$

$$IDM = \sum_{i=1}^{L}\sum_{j=1}^{L} \frac{P(i,j)}{1+(i-j)^2} \quad (2)$$

$$Energy = \sum_{i}\sum_{j} P^2(i,j) \quad (3)$$

$$Entropy = \sum_{i=1}^{L}\sum_{j=1}^{L} P(i,j).\log\left(P(i,j)\right) \quad (4)$$

$$Homogeneity = \sum_{i}\sum_{j} \frac{1}{1+(i-j)^2} P(i,j) \quad (5)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Yi - \hat{Y}i\right)^2 \quad (6)$$

On the other hand, HOG describes the appearance of the object by a gradient or edge intensity distribution. The features obtained from the HOG are calculated by taking an edge-oriented histogram in the local area. The magnitude and direction of HOG is calculated using the following equation.

$$Magnitude = \sqrt{gx^2 + gy^2} \quad (7)$$

$$Direction = arctan\frac{gy}{gx} \quad (8)$$

These combined features are then fed to the Support Vector Machine (SVM) [12] to be classified into three classes, namely organic trash, non-organic trash and not a trash. The parameters setting for the SVM are shown in Table 1.

To analyze the performance, an analysis is carried out by looking for accuracy, precision, recall and F-1 score. The following is the formula used to get the system's performance:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

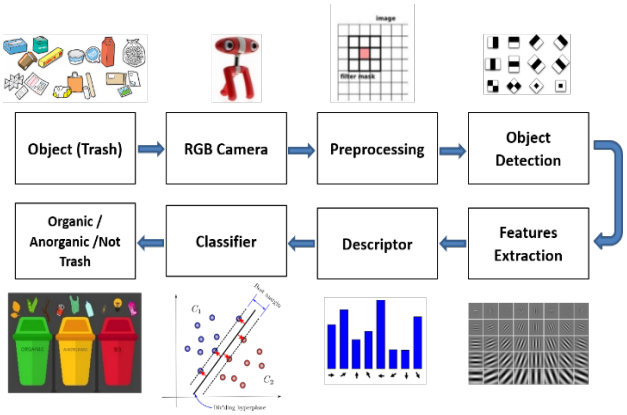


Fig 1. Trash detection and classification design

Table 1. The parameters setting for the SVM

Parameters	Value
Type	C-SVC
Kernel type	Linear

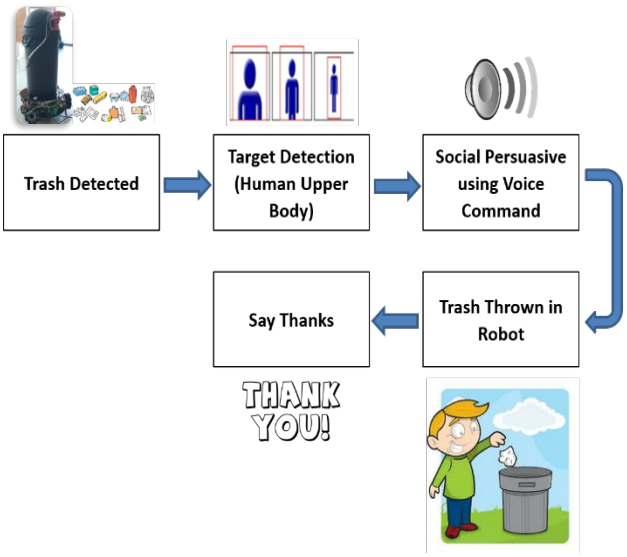


Fig 2. Target detection and social persuasive design

$$F1Score = 2x\frac{Precision * Recall}{Precision + Recall} \quad (12)$$

B. *Human Target Detection and Social Persuasive*

Apart from being able to detect and distinguish types of trash, ATRACBOT also has the ability to detect human existence in the surrounding environment. Human detection is needed because the robot that we have developed is a social robot, where interaction with humans is one of the main targets. Here, the role of robots is to teach humans about the importance of disposing of waste in its place. Figure 2 shows the system's design of the human target and social persuasive detection system.

The working of this system is that when the robot successfully detects a trash object, the camera installed on the robot will perform the second function, namely looking for human targets that have the potential to help the robot put trash into the trash bin provided above the robot. Human detection is carried out on the upper part of the human body based on shape features using the HOG and SVM methods [4]. When someone is detected, the robot will make a sound to invite the person to take and put the garbage in the trash bin. If the person, does it, then the robot will sound again saying thank you. If it has not been done, the robot will continue to sound to invite the person. The previous research has been done using a single core CPU programming and obtains detection results with an accuracy of 71% and fps of 4.26 fps.

C. *Robot Operating System (ROS)*

ROS is an open-source Robot Operating System in which there are libraries and tools for creating robot software. ROS is a robotic middleware that can flexibly connect robotic hardware to a computer operating system. ROS aims to make it easier for robot developers to create their software without having to create source code from scratch and can be developed together.

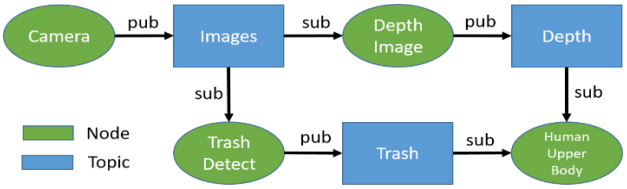


Fig 3. Node design

ROS has 3 concepts, namely the filesystem level, the computational graphics level and the community level[13] [14]. The filesystem level is the level of ROS resources available on the system. the first step when using or developing ROS is to conceptualize it like an operating system. In an ROS filesystem there are folders, and each folder has a different file description according to its respective function.

Figure 3 shows the node design used in this system. This system consists of 4 nodes and 3 topics that are created. The publish subscribe model is used so that data that has been processed on a node can be used simultaneously by other nodes. Camera node to take pictures of the environment using a minoru stereo camera. Then obtained data in the form of images that are processed in the depth image node so as to produce a depth image that has silhouette image data and the distance of the object from the camera. The depth image is used to detect human presence using the human upper body method, which is created by a separate node. Social persuasion is combined by using the human upper body node which will run when a human is detected.

Figure 4 shows a flowchart to explain the workflow of the robot. The first step is to initialize the camera, sensors

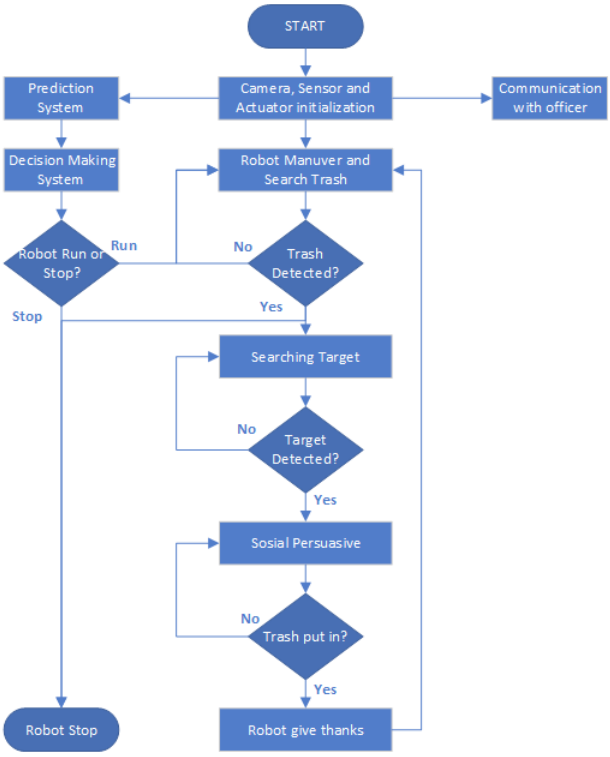


Fig 4. Flowchart robot

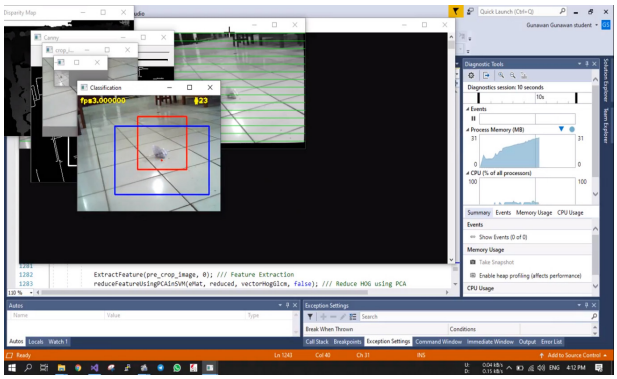
and actuators. Then the robot will maneuver looking for trash at a slow speed so that it does not endanger the surrounding environment. After the trash is detected, the robot will stop and search for human targets to dispose of the detected trash into the space provided to the robot. To detect human targets around the robot, the human upper body detection method is used. After the target is detected, the robot will attract the target's attention to dispose of the trash around the robot. Then after the trash is entered, the robot will give appreciation to the person who has entered the trash. Then the robot will maneuver to find trash again.

Prediction systems and decision-making systems are used to provide intelligence to robots that run autonomously to be able to make decisions independently. Meanwhile, communication with officers is a feature used to monitor robots using the telegram application remotely.

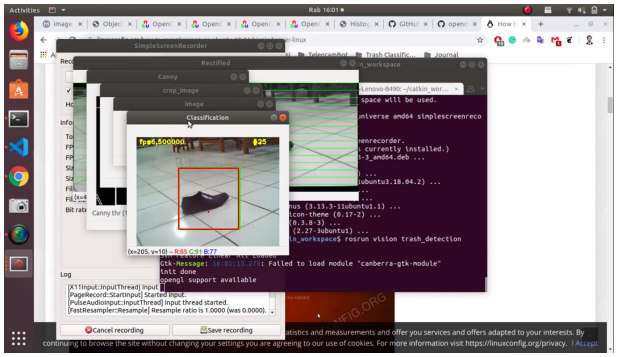
IV. Result and Discussion

To perform system testing, testing is carried out using two methods, namely running programs on different operating systems and comparing with running programs on ROS. The operating systems being compared are Visual Studio 2017 on Windows 10 with ROS Melodic on Ubuntu 18.04 LTS. For the test location, data collection is made the same so as to minimize environmental differences.

Figure 5 shows a trash detection and classification system using the ROS platform. Where there are two nodes, namely stereo publisher and trash detection as shown in Fig. 7 which is generated using rotograph on ROS. The

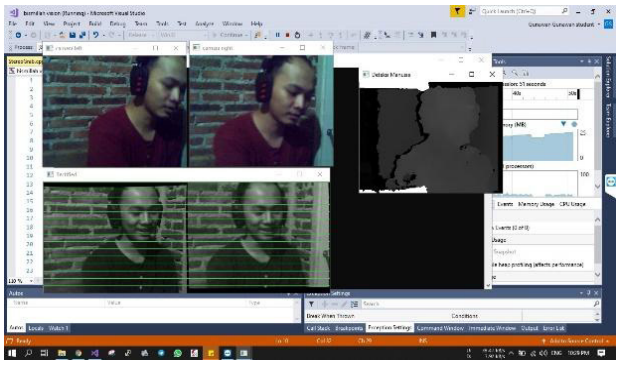


(a) Trash detection run in windows

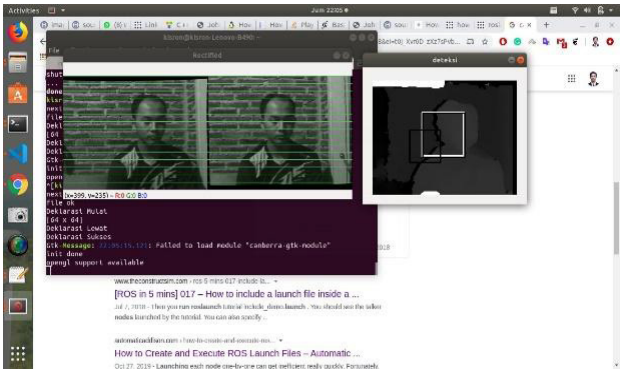


(b) Trash detection run in linux

Fig 5. Running trash detection program



(a) Target detection run in windows



(b) Target detection run in linux

Fig 6. Running target detection and social persuasive program

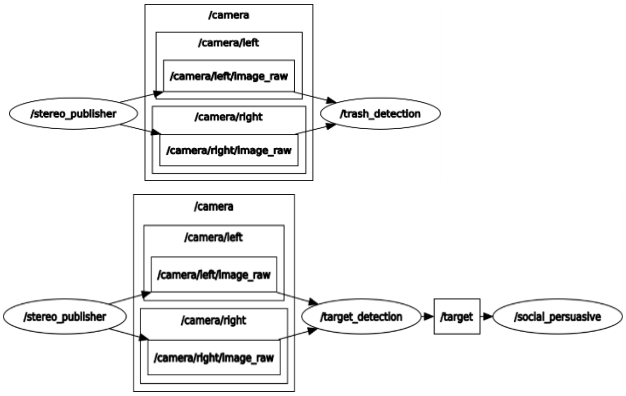


Fig 7. ROS Node graph

two separate nodes will reduce the computational load that runs sequentially, the time needed to update the data capture image will also be cut quite a lot so that the detection process will be faster. After converting from windows to ROS, the initial fps increase was 3.221 to 7.003.

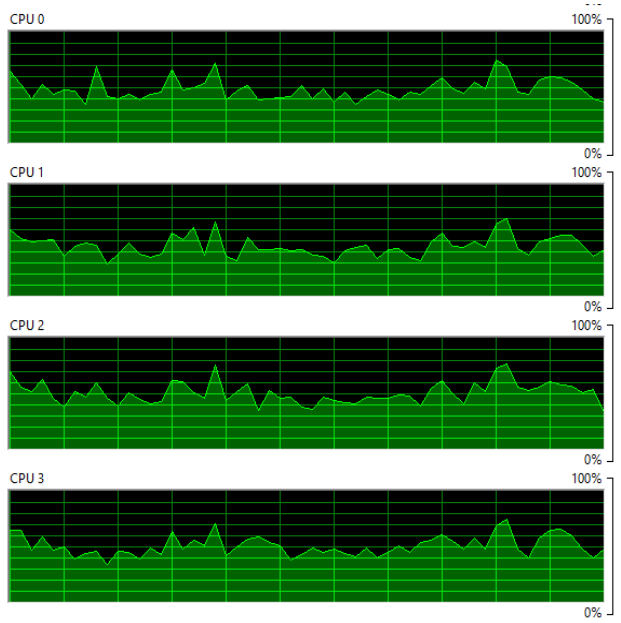
Figure 6 shows a trash detection and classification system using the ROS platform. Where there are three nodes, namely stereo publisher, target detection and social persuasive as shown in Fig. 7 which is generated using rosgraph on ROS. The three separate nodes will reduce the computational load that runs sequentially, the time needed to update the data capture image will also be cut quite a lot so that the detection process will be faster. The social persuasive node will run when the target detection

Table 2. FPS comparison

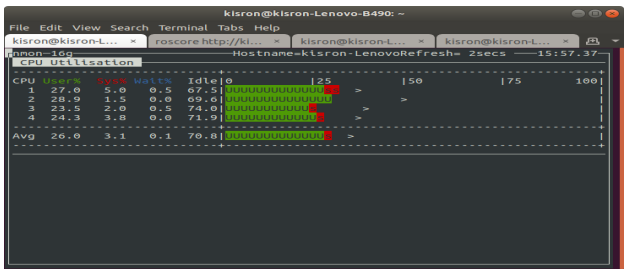
Running on	Trash Detection (fps)	Target Detection (fps)
Windows	3.221	4.442
Linux (ROS)	7.003	8.515

node successfully detects human presence around the robot. After converting from windows to ROS, the initial fps increase was 4,442 to 8,515. From the results of the experiments that have been carried out, it can be summarized into tables to make it easier to analyze data. Table 2 shows the results obtained.

FPS or frames per second is the number of images/ frames displayed in 1 second [15]. The greater the fps of a video, the smoother and smoother the image displayed is. And the lower the fps, the worse the video quality. Based on Table 2, the results of running programs on Linux have higher fps when compared to running programs on Windows. This is because when running the program in Windows, the distribution of the CPU cores is not evenly distributed so that the program is run sequentially. The high and low FPS is influenced by the processes run by the system. The previous systems run sequentially and do not involve all CPU cores in the process. The system uses ROS which divides each CPU core to work on the process



(a) Task Manager Windows



(b) Task Manager Linux

Fig 8. CPU Performance Comparison

Table 3. Confusion matrix trash detection offline testing (Salimi [3])

		Actual Values	
n=100		Positive	Negative
Predicted Values	Positive	TP = 67	FP = 12
	Negative	FN = 14	TN = 7

Table 4. Confusion matrix target detection offline testing (Dewantara [4])

		Actual Values	
n=100		Positive	Negative
Predicted Values	Positive	TP = 67	FP = 12
	Negative	FN = 13	TN = 8

so that the system becomes multi-threaded. This causes the FPS to be better.

Figure. 8 (a) shows the CPU performance in running the program. Of the four cores, it can be seen that CPU usage is above 50%. This makes the computer work hard to run the program. Meanwhile, using ROS on Linux, the CPU performance shown in Fig. 8 (b) has an average value of 26%. So that with that much CPU usage it doesn't burden the work of the computer itself.

In addition to the reduction in processor workload, the detection system performance is tested using parameters accuracy, precision, recall and F-score by testing offline and real implementation. Testing offline by detecting images 100 times with different objects and positions. The real implementation is executed by enabling detection for 60 seconds and replacing it with a different object. Table 3 shows the offline trial of trash detection using previous research method [3] and then Table 4 shows the offline trial of target detection using previous research [4]. Then, Table 5 and Table 6 shows offline trial trash and target detection using our method. Table 7 shows the result of real implementation using previous method [3] and table 8 shows the result of real implementation using previous method [4]. Then, Table 9 and Table 10 show the result of real implementation trash and target detection using our method.

The offline experiments were carried out by detecting 100 times the detection of trash and detection of targets.

Table 5. Confusion matrix trash detection offline testing (ours)

		Actual Values	
n=100		Positive	Negative
Predicted Values	Positive	TP = 72	FP = 10
	Negative	FN = 13	TN = 5

Table 6. Confusion matrix target detection offline testing (ours)

		Actual Values	
n=100		Positive	Negative
Predicted Values	Positive	TP =75	FP = 7
	Negative	FN = 12	TN = 6

Table 7. Confusion matrix trash detection for real implementation (Salimi [3])

		Actual Values	
n=100		Positive	Negative
Predicted Values	Positive	TP =60	FP = 15
	Negative	FN = 13	TN = 22

Table 8. Confusion matrix target detection for real implementation (Dewantara [4])

		Actual Values	
n=100		Positive	Negative
Predicted Values	Positive	TP =79	FP = 18
	Negative	FN = 23	TN = 25

Table 9. Confusion matrix trash detection for real implementation (ours)

		Actual Values	
n=100		Positive	Negative
Predicted Values	Positive	TP =168	FP = 37
	Negative	FN = 51	TN = 82

The results obtained on the trash detection system accuracy 77,00%, precision 87,80%, recall 82,75%, and F1 score 85,20%, while the target detection results in performance accuracy 81%, precision 91,46%, recall 86,20%, and F1 score 88,42 %. Table 7 shows the performance comparison based on accuracy, precision, recall and F1 score.

In real implementation there are differences in the frames that are processed. by using ROS frames that are processed more than using the previous researches. Table 11, 12, 13, 14, 15 and 16 show the detail of trash and target detection for each experiment. Where from the results of the tests we did, the technique we used had a better performance than the results in [3] and [4].

From the result of the experiments above, we can observe that there is an increase in each parameter. This is directly proportional to the reduction in processor workload, thereby increasing detection results. The combination of HOG and GLCM features is also considered to be able to improve system performance. Research by Gue and Liu [16] proved that the combination of HOG and GLCM

Table 10. Confusion matrix target detection for real implementation (ours)

		Actual Values	
n=100		Positive	Negative
Predicted Values	Positive	TP =94	FP = 30
	Negative	FN = 25	TN = 48

Table 11. Detail trash detection offline testing result (Salimi [3])

Trash Detection		Detection			Accuracy
		Organic	Non-Organic	Not Trash	
Actual	Organic	25	7	3	71,42%
	Non-Organic	4	26	5	74,28%
	Not Trash	2	5	23	76,67%
n=100				Average	74,12%

Table 12. Detail trash detection offline testing result (ours)

Trash Detection		Detection			Accuracy
		Organic	Non-Organic	Not Trash	
Actual	Organic	**26**	6	3	74,28%
	Non-Organic	4	**29**	2	82,85%
	Not Trash	2	4	**24**	80,00%
n=100				Average	93,04%

Table 13. Detail trash detection for real implementation result (Salimi [3])

Trash Detection		Detection			Accuracy
		Organic	Non-Organic	Not Trash	
Actual	Organic	**58**	13	9	72,50%
	Non-Organic	10	**56**	9	74,67%
	Not Trash	3	5	**22**	73,33%
n=185				Average	73,50%

Table 14. Detail trash detection for real implementation result (ours)

Trash Detection		Detection			Accuracy
		Organic	Non-Organic	Not Trash	
Actual	Organic	**147**	31	22	73,50%
	Non-Organic	24	**137**	20	75,69%
	Not Trash	12	20	**93**	74,40%
n=506				Average	74,53%

Table 15. Performance comparison of offline testing

Performance Indicator	Off Trash Detection (%)		Human Target Detection (%)	
	Salimi [3]	Ours	Dewantara [4]	Ours
Accuracy	74,00%	**77,00%**	75,00%	**81,00%**
Precision	84,81%	**87,80%**	84,81%	**91,46%**
Recall	82,71%	**82,75%**	83,75%	**86,20%**
F1 Score	83,75%	**85,20%**	84,27%	**88,42%**

Table 16. Performance comparison of real implementation

Performance Indicator	Real Trash Detection (%)		Human Target Detection (%)	
	Salimi [3]	Ours	Dewantara [4]	Ours
Accuracy	**74,54%**	73,96%	71,72%	**72,08%**
Precision	80,00%	**81,95%**	**81,44%**	75,80%
Recall	82,19%	76,71%	77,45%	**78,99%**
F1 Score	**81,08%**	79,24%	**79,39%**	77,36%

features from images then fed into SVM for experiments demonstrated effectiveness. The combination of the two in image classification is superior to the use of HOG alone or GLCM alone.

V. Conclusion

After evaluating system performance and integration, it can be concluded that by using the publish-subscribe method on ROS Performance, the trash detection system can run at a speed of 7,003 fps. Meanwhile, the target detection system is capable of running at a speed of 8,515 fps. In addition to this increase, CPU usage has also decreased, which originally with a single process using CPU resources is quite large with an average of 52% to 26% only. This of course will maintain the stability of system performance and also the devices used and can use one camera to run two systems simultaneously. In addition to a decrease in CPU usage, there was an increase in the performance of each system which was able to achieve 77% accuracy, 87.80% precision, 82.75% recall and 85.20% F1 Score in the trash detection and classification system. Then the detection system for human and social persuasive targets has increased in accuracy reaching 81%, precision reaching 91.46%, recall reaching 86.20% and F1 Score reaching 88.42%.

The next task is to integrate with communication systems and navigation systems. Then after integration, the robot will run in a real environment, namely in the form of public facilities. Then it can be evaluated the robot's performance based on the results obtained.

Reference

[1] Badan Pusat Statistik, "Laporan Indeks Perilaku Ketidakpedulian lingkungan hidup indonesia 2018," pp. 44, 2018, doi: 978-602-432-210-0.

[2] BPS, "Badan Pusat Statistik: Statistical Yearbook of Indonesia 2018, accessed: www.bps.go.id.," 2018.

[3] I. Salimi, B. S. B. Dewantara and I. K. Wibowo, "Visual-based trash detection and classification system for smart trash bin robot," 2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC), 2018, pp. 378-383, doi: 10.1109/KCIC.2018.8628499.

[4] B. S. B. Dewantara, F. Ardilla and A. A. Thoriqy, "Implementation of Depth-HOG based Human Upper Body Detection On A Mini PC Using A Low Cost Stereo Camera," 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), 2019, pp. 458-463, doi: 10.1109/ICAIIT.2019.8834580.

[5] Kisron, B. S. B. Dewantara and F. Ardilla, "Self Monitoring, Failure-Detection and Decision-Making System to Support E-TrashBot (EEPIS Trash Bin Robot) Operations: Preliminary Report," 2018 10th International Conference on Information Technology and Electrical Engineering (ICITEE), 2018, pp. 1-6, doi: 10.1109/ICITEED.2018.8534932.

[6] Kisron, B. S. B. Dewantara and F. Ardilla, "Early Warning and IoT-based Reporting System for Mobile Trash Bin Robot Application," 2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC), 2018, pp. 341-348, doi: 10.1109/KCIC.2018.8628550.

[7] F. A. Haq, B. S. B. Dewantara and B. S. Marta, "Room Mapping using Ultrasonic Range Sensor on the ATRACBOT (Autonomous Trash Can Robot): A Simulation Approach," 2020 International Electronics Symposium (IES), 2020, pp. 265-270, doi: 10.1109/IES50839.2020.9231734.

[8] A. Roy *et al.*, "Performance comparison of machine learning platforms," *INFORMS Journal on Computing*, vol. 31, no. 2, pp. 207–225, 2019, doi: 10.1287/ijoc.2018.0825.

[9] A. Ashari, I. Paryudi, and A. Min, "Performance comparison between naïve bayes, decision tree and k-nearest neighbor in searching alternative design in an energy simulation tool," *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 11, pp. 33–39, 2013, doi: 10.14569/ijacsa.2013.041105.

[10] A. Kika, "Multithreading image processing in single-core and multi-core CPU using java," *International Journal of Advanced Computer Science and Applications*, vol. 4. No. 9, pp. 165-169, 2013, doi: 10.14569/IJACSA.2013.040926.

[11] S. Al-Aidid and D. Pamungkas, "Sistem pengenalan wajah dengan algoritma haar cascade dan local binary pattern histogram," *Jurnal Rekayasa Elektrika*, vol. 14, no. 1, pp. 62–67, 2018, doi: 10.17529/jre.v14i1.9799.

[12] C.C. Chang and C.J. Lin, "LIBSVM: a library for support vector machines", ACM Trans. Intelligent System Technology, Vol. 2, No. 27, pp. 1–27, 2011.

[13] A. Martinez and E. Fenandez. 2013. Learning ROS for Robotics Programming. Birmingham, UK: Packt Publishing.

[14] E. Fernandez, L. S. Crespo, A. Mahtani and A. Martinez. 2015. Learning ROS for Robotics Programming - Second Edition. Birmingham, UK: Packt Publishing

[15] A. Vyas, S. Yu, and J. Paik. 2018. Fundamentals of Digital Image Processing. In: Multiscale Transforms with Application to Image Processing. Signals and Communication Technology. Springer, Singapore. https://doi.org/10.1007/978-981-10-7272-7_1.

[16] G. Jianyue and L. Haoting. 2020. Investigation of Image Classification Using HOG, GLCM Features, and SVM Classifier. In: Long S., Dhillon B.S. (eds) Man-Machine-Environment System Engineering. MMESE 2020. Lecture Notes in Electrical Engineering, vol 645. Springer, Singapore. https://doi.org/10.1007/978-981-15-6978-4_49.