Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th February 2025. Vol. 4. No. 2; e-ISSN: 2808-4519

Automatic Waste Type Detection Using YOLO for Waste Management Efficiency

Alfattah Atalarais[1]*, Kana Saputra S[2], Hermawan Syahputra[3], Said Iskandar Al Idrus[4], Insan taufik[5]

[1,2,3,4]*Ilmu Komputer, Universitas Negeri Medan*
alfatahatalarais12@gmail.com[1]*

Abstract

The management of waste in Indonesia is currently suboptimal, with only 66.24% being effectively managed, leaving 33.76% unmanaged. This highlights a significant challenge in waste management, primarily due to a lack of understanding in selecting appropriate waste types. Advances in deep learning and computer vision offer promising solutions to this issue. This study employs the YOLOv8l model, a well-regarded deep learning model for object detection, to develop an automated waste type detection system integrated with trash bins. The dataset comprises 2800 images across four classes, each containing 700 images, and is split with an 80:10:5 ratio for training, validation, and testing. Evaluation on test data yields a mean Average Precision (mAP) of 96.8%, indicating robust model performance in object detection. The model's accuracy is further validated with a score of 89.98%. Real-time testing conducted at Merdeka Park, Binjai, demonstrates the system's capability to detect waste with varying confidence levels, consistently above the 0.5 threshold. The highest confidence was observed in bottle detection at 0.94, and the lowest in cans at 0.64, underscoring the system's reliability across different detection scenarios within a 30cm range.

Keywords: *Waste management, Deep learning, Detection system, Real-time detection, YOLOv8*

1. Introduction

The waste problem in Indonesia has reached an alarming level. Based on data from the Ministry of Environment and Forestry, waste generation in Indonesia reaches 25 million tons per year, with 33.7% of which is not managed properly [1] this can be seen in Figure 1.

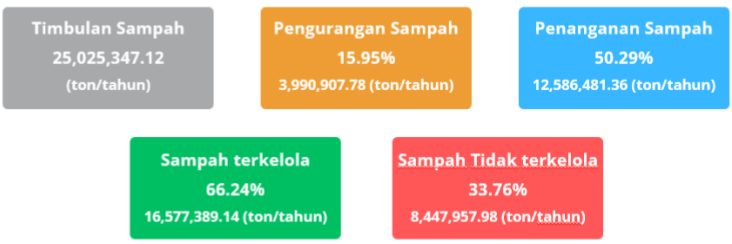

Fig. 1: Data on waste management achievements in Indonesia

Based on the figure above, it can be seen that the achievement of waste management in Indonesia still requires serious attention. Well-managed waste only reached 66.24%, while unmanaged waste still stands at 33.76%. This data shows that a third of all waste in Indonesia has not been handled effectively, highlighting the urgent need for innovation and improvement in the waste management system to maintain environmental hygiene and public health.

The problem of waste management in Indonesia, especially in North Sumatra Province, is a significant problem [2] Based on data from the National Waste Management Information System [3] the total waste generation in North Sumatra in 2023 is estimated to reach 1.1 million tons. The composition of waste in North Sumatra consists of 60% organic waste, 12.5% plastic waste, 17.16% paper waste, 3.64% fabric waste, 4.24% glass waste, 1.86% metal waste, and 15.55% other waste. This can be seen figure 2.

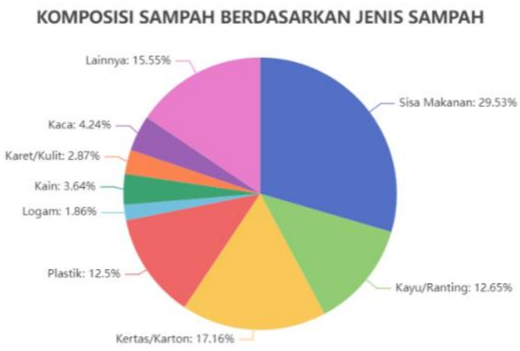

Fig. 2: Composition of waste types in North Sumatra

Based on the data that has been presented, we can see that the percentage of waste composition in other categories is still relatively large, which is 15.55%. Therefore, we can understand that one of the main obstacles in waste management in the community is the lack of understanding about the correct waste selection. The problem of waste disposal that does not fit its category covers various aspects, including human behavior and technological challenges. A major challenge is the lack of individual awareness and understanding of the type of waste that should be placed in each bin, which results in improper disposal and undermines the purpose of the waste management system. To overcome this problem, more effective efforts are needed in separating and managing waste, one of which is by utilizing technology [4].

Technology plays an important role in supporting efforts to tackle Indonesia's waste problem. With technological advancements, we can find new ways to manage waste more efficiently and sustainably. In response to these challenges, developments in deep learning and computer vision technologies offer promising solutions. One popular and effective deep learning model for object detection tasks is YOLO (You Only Look Once). Automatic detection technology of waste types using the YOLO (You Only Look Once) method can be an innovative solution in improving waste management efficiency. YOLO is one of the deep learning algorithms that can perform real-time object detection with high accuracy. Research on litter detection using YOLO (You Only Look Once) has shown significant progress in improving the accuracy and efficiency of waste management systems. Various studies have explored different versions and improvements of the YOLO algorithm to address specific challenges in litter detection. Implementation of the YOLO model in urban areas has proven high accuracy in detecting overflowing bins, with YOLOv5n achieving an accuracy rate of 94.50% [5]. Then in the context of medical waste management in identifying surgical waste such as masks and gloves the role of object detection using YOLO is very helpful in reducing the risk of infection and increasing the efficiency of waste sorting [6]. Then research by [7] where using the Skip-YOLO model overcomes the challenges of recognizing waste through the expansion of the receptive field and the integration of high-dimensional multiscale feature maps increases detection accuracy by 22.5% compared to YOLOv3.

The use of automatic detection technology for waste sorting using YOLO (You Only Look Once) is expected to provide an effective and efficient solution in overcoming waste management problems in the community. Given the high percentage of other categories of waste composition at 15.55%, as well as the main obstacles that include a lack of understanding of waste segregation and low environmental awareness, the implementation of this detection system can improve the efficiency of real-time waste segregation. The detection system is expected to help people understand the difference between organic and inorganic waste better, enabling more effective waste management. In addition, this technology is also expected to be a valuable source of information for authorities in evaluating and designing better waste management strategies in the future. Thus, the waste type detection system using YOLO not only aims to improve the efficiency of waste management, but also to increase public awareness about waste segregation, protect the environment, and create a more sustainable waste management system. Therefore, this research aims to explore the potential of applying YOLO in waste type detection as an innovative solution in waste management.

2. Research Method

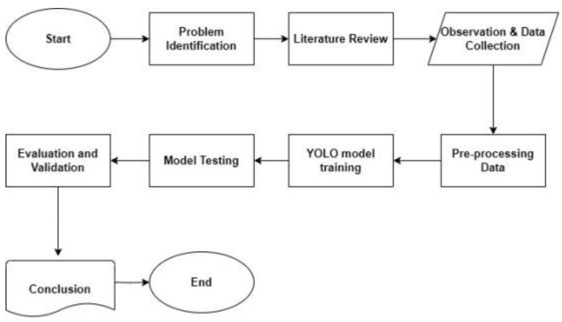

Fig. 3: Research Workflow

2.1. Problem Identification

In the early stages of this research, researchers identified the main problem that became the background of the study. At this stage, researchers found a problem in the form of a lack of awareness and knowledge of the community in sorting organic and inorganic waste. This causes waste to be poorly managed, so an automated system is needed that can help detect the type of waste before disposal.

2.2. Literature Review

At this stage, researchers conducted a literature review to understand the concepts, theories, and technologies relevant to the research to be carried out. The literature study includes searching scientific articles, journals, and other academic sources that discuss waste detection systems, YOLO (You Only Look Once) technology and the application of computer vision technology in waste management. The purpose of the literature study is to obtain a strong theoretical foundation and identify research gaps that will be filled by this research.

2.3. Observation & Data Collection

At this stage, researchers collect data directly to build the primary dataset. The data obtained is in the form of images of general waste, such as plastic bottles, beverage cans, orange peels, and eggshells taken using a camera that will be applied in the system. This data collection process aims to complement the secondary dataset from Roboverse Universe, so as to improve the accuracy of garbage detection and classification by the YOLO algorithm. The resulting dataset will be used in the development and testing process of the automatic waste detection system, to ensure that the system can effectively recognize the type of waste in real conditions.

2.4. Pre-processing Data

After collecting the dataset of junk images, a data preprocessing stage is performed to improve the input quality for the YOLO algorithm. This stage is very important to reduce the possibility of errors in the image and optimize detection accuracy. The preprocessing process begins with labeling where each trash image is labeled according to its category (Bottles, Cans, Orange peels and Eggshells) for class identification in the classification process. Next, the images are converted to a uniform size, e.g. 640x640 pixels to ensure input consistency and efficiency in computation. Data augmentation techniques are also applied to enrich the dataset by generating variations of the existing images, such as rotation, brightness adjustment, or noise addition. This step aims to allow the model to learn more patterns and increase robustness to variations in real-world conditions. Finally, the preprocessed data is divided into training and validation data, a crucial step to ensure the model's ability to generalize well and avoid overfitting. With this proper preprocessing step, it is expected that the YOLO algorithm can work more efficiently and provide more accurate detection results in classifying waste into organic or inorganic.

2.5. YOLO Model Training

The model training process is carried out using data that has been labeled from the training data set. The main purpose of this training is to ensure that the YOLO model can learn and recognize specific patterns and features of organic (Orange peel & Eggshell) and inorganic (Plastic bottle & Beverage can) waste according to the research objectives. The training data is used to hone the model's ability to identify and classify the types of waste accurately and effectively. The model is trained through several iterations until it reaches the desired level of accuracy.

2.6. Model Testing

After the YOLO model is trained, the next stage is testing the model to evaluate its performance. This testing is done using test data that is different from the training data to ensure that the model can generalize well to new data. The testing process involves applying the model to the test data to detect and classify the type of waste. The detection results are compared with the correct labels to calculate performance metrics such as accuracy, precision, and recall. This testing is important to assess how well the model can perform litter detection in real conditions.

2.7. Evaluation & Validation

Evaluation and validation are critical steps to ensure that the trained and tested model works as expected. At this stage, the results from model testing are analyzed to determine the strengths and weaknesses of the model. Performance metrics such as accuracy, precision, and recall are calculated to provide an overall picture of the model's performance. Validation is done by comparing the model's predicted results with the actual data to ensure the model has high reliability. If required, the model can be refined by retraining or adjusting parameters to achieve better performance.

2.8. Conclusion

Drawing conclusions based on the results of research and testing that has been carried out by researchers.

3. Results and discussion

3.1. System design

This research aims to design a system by utilizing programming languages and hardware components selected based on research needs. The main objective of the research is to build a smart trash can system by utilizing object detection from YOLOv8 technology which is integrated into the Arduino UNO microcontroller to open and close the trash can according to the type of garbage by the servo motor. The hardware and software components used in this research, including:

Table 1: Specification hardware & software

Hardware	Spesification	
Laptop	• Processor Intel® Core™ i5- i5-8300H @2.30GHz with Intel UHD Graphics 630 • 8GB single-channel RAM	• NVIDIA GeForce GTX 1050 GPU 4GB GDDR5 • SSD SATA 1TB TeamForce EX2 & SSD Nvme 512GB TeamForce MP33
Smartphone	• Samsung Galaxy A52s • 64 megapixel camera	• 8GB of memory • 256GB of storage
Arduino UNO	• Atmega328 SMD • 5v operating voltage • 7-12v input voltage • 14 digital I/O pins • 6 Analog inputs pins	• 32kb Flash memory • 2kb of SRAM • 1kb EEPROM • 1MHz of Clock speed
Servo	• Towerpro Motor Servo SG90 9G • 180° of servo rotation • 1,8kg/cm(4,8v) of stall torque	• 0,1sec/60 of degree of operating speed • 4,8v operating voltage • 25cm of length wire
Software		
	• Windows 11 64-bit • Visual Studio Code 1.93.1 • Python 3.12.5 • Google Collab	• Arduino IDE 2.3.3 • Droidcam 6.5.2 • Brave Browser 1.73.89 • Roboflow

3.2. Data Collection

This research uses a dataset consisting of 2800 waste images that are evenly distributed into four categories, namely bottles, beverage cans, orange peels, and eggshells with each category having 700 image samples. In this research, there are two methods of image data collection, namely primary data collection and secondary data. Primary data is collected through direct shooting conducted by researchers while secondary data collection comes from public or open source datasets that have been provided in Roboflow.

In the primary data collection process, the researcher tried to ensure the diversity of the dataset by photographing the waste from different angles, lighting conditions, and locations. Each class in the dataset has unique characteristics identified by shape, color, and texture. The characteristics of each object can be seen in table 2 below:

Table 2: Sample characteristics

Characteristics / Object	Shape	Color	Texture	Retrieval Method
Plastic Bottle	Cylinder-shaped with a narrower neck	Transparent or colored	Smooth/slightly bumpy surface with brand label present	Front and side views of the object
Drink cans	Cylinder-shaped with flat top and bottom surfaces	Metallic/aluminum with colored labels	Shiny and hard surface	Front and side views of the object
Orange peels	Roughly textured, slightly curved, tends to be irregular	Bright Yellow/Orange	Rough and porous	Front and side views of the object
Eggshells	Flat, slightly curved, tends to be irregular	Brown	Brittle and hard	Front and side views of the object

This combination method is expected to improve the accuracy of litter detection and classification in various environmental conditions, improving the system's ability to recognize litter objects. Here are some examples of dataset retrieval that have been done:

Fig. 4: Example of dataset image capture

3.3. Pre-processing Data

Before starting the model training process using the data that has been collected, it is very important to perform a data preprocessing stage. This step aims to ensure that the data to be used in the training process has been optimized so that the system can detect different types of waste more accurately and efficiently. In the context of machine learning, data preprocessing involves several crucial stages, including the selection of high-quality data, proper and consistent data labeling, division of data into training, validation, and testing sets, adjustment of image size for uniformity, and data augmentation to enrich the variety of datasets. At this stage, we use Roboflow as a place to perform data preprocessing.

3.3.1. Dataset labeling and division

The dataset that has been collected at the data collection stage is then subjected to a labeling process on each image. This labeling process is done manually and one by one on the image of each class. It can be seen in Figure 5, where the labeling is done by segmenting the area using the smart polygon feature that is available in Roboflow on the specific object that you want to detect, which in this study are bottles, cans, orange peels and eggshells to ensure that the model only focuses on relevant objects during training.

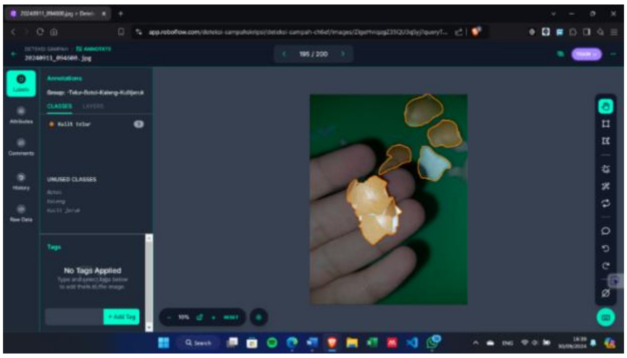

Fig. 5: Labelling Image Data on Roboflow

After doing the image labeling stage on the dataset, the next step is to divide the entire dataset into 3 categories, namely training, validation and test. The data is divided with a ratio of 85:10:5, where 85% is used as training data, 10% as validation data and 5% as test data. The purpose of dividing this data into 3 categories is so that the model can be trained with sufficient data, validated to prevent overfitting and ensure the generalization ability of the model and tested to evaluate the final performance of the model in detecting litter in situations that resemble real conditions.

3.3.2. Image Risizing and Augmentation

In the model training process, the risizing and augmentation stages of the image are quite important to generate visual information consistently by uniforming the input size and enriching the training dataset with the aim of improving the efficiency and accuracy of model training, as well as optimizing memory usage during the learning process of the YOLOv8 model. In this study, we resized or risized the image to 640×640 resolution. This resolution was chosen because it provides the best balance between image detail and computational efficiency. This size has become a popular standard in modern object detection models, especially in the YOLO architecture which is often used for real-time object detection. This resolution is large enough to capture small objects such as bottles, cans, orange peels and eggshells as well as the necessary visual context.

After risizing the image, the next step is data augmentation. Augmentation is a method used to increase the amount and variety of training data through modifying existing images. This stage is very important, especially when the number of available datasets is limited, as it can help improve the performance of the model in the training process. With augmentation, we can enrich the dataset without having to collect more new images while improving the model's ability to recognize litter objects under various conditions that may be encountered in the field. The augmentation techniques that the researcher applied to the study include flip, crop, shear, blurring, noising, rotation, as well as saturation, brightness, and exposure adjustments.

Fig. 6: Image Augmentation Example

3.4. Training Model

After the data has gone through the preprocessing stage, the data will then enter the model training stage. In this study, researchers used the YOLOv8 Architecture for the object detection process. YOLOv8 has 5 model variants, each of which has a different size and complexity. Each variant is designed to balance processing speed and detection accuracy according to project requirements and computational constraints. The selection of the right variant also depends on the complexity of the dataset, the objectives and the available computing infrastructure.
There are variants of YOLOv8, namely nano(n), small(s), medium(m), large(l) and extra large(xl). It can be seen in the performance comparison table of each model variant, that the larger the size of the model used, it requires a long processing time but is directly proportional to the resulting accuracy.

Table 3 YOLOv8 Model Variant Comparison

Model	Size	mAPval 50-95	Speed CPU ONNX	Speed A100 TensorRT	Params	FLOPs
YOLOV8n	640	37.3	80.4	0.99	3.2	8.7
YOLOV8s	640	44.9	128.4	1.20	11.2	28.6
YOLOV8m	640	50.2	234.7	1.83	25.9	78.9
YOLOV8l	640	52.9	375.2	2.39	43.7	165.2
YOLOV8xl	640	53.9	479.1	3.53	68.2	257.8

In this study, we used YOLOv8l to train the model by utilizing Google Collab as a model training platform. In the process, we also use the additional L4 GPU that is available in Google Collab Pro to help speed up the model training process. Before starting the model training process, researchers first ensured that the integration between Google Collab and Google Drive was properly configured to access and store training data and ensure the availability of the necessary libraries in the Google Collab environment. Before starting the model training process, researchers first ensured that the integration between Google Collab and Google Drive was well configured to access and store training data and ensure the availability of the necessary libraries in the Google Collab environment.

After ensuring that the entire environment or training environment on Google Collab has been fulfilled as well as the location of the appropriate dataset, the next step is to initialize the model that will be used in training by adjusting several hyperparamater functions in YOLOv8. Hyperparamater that researchers adjust include the number of epochs, batch size, and image size. The table below shows the value of each hyperparameter in the YOLOv8l model used for training which aims to produce good training data results.

Table 4 Training Hyperparameter Settings

Hyperparameter	Value
Epoch	100
Batch size	16
Image size	340

After initializing the YOLOv8 hyperparameter function, the next step is to start the training process on the model.

3.5. Training results

During the training process, YOLOv8 learns and optimizes its parameters based on the patterns in the training dataset. This learning process allows the model to develop powerful generalization capabilities. Ultimately, this results in a trained YOLOv8 model that is able to perform accurate object detection on new data that has never been seen before, reflecting the effectiveness of YOLOv8 in capturing and utilizing key information from the training data for real-time object detection applications. The results of the training and validation process of the YOLOv8l model using the pre-defined hyperparameters are shown in Figure 7.

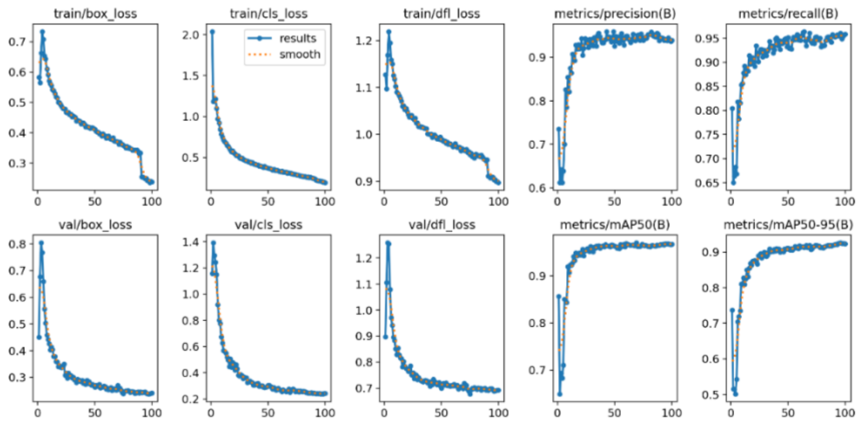

Fig. 7: YOLOv8l Model Training Results Graph

The parameter graph of the model training results can be seen in Figure 7, in the first row of the graph, the downward trend that occurs can be said to be quite consistent for train/box_loss, train/cls_loss, and train/dfl_loss, indicating that the model gradually improves its ability to predict object locations (box_loss), classify objects correctly (cls_loss) and refine the estimation of object distributions (dfl_loss). This steady decrease in loss indicates that the model learns effectively from the training data. The second row of graphs displays similar losses for the validation data, namely val/box_loss, val/cls_loss, and val/dfl_loss. The similar pattern of decrease between the training and validation data indicates that the model does not suffer from significant overfitting. This means that the model is able to generalize well to data that has never been seen before. The evaluation metrics in the last two graphs in the first row, namely metrics/precision(B) and metrics/recall(B) show a steady increase and reach high values. This indicates that the model has high accuracy in detecting objects (precision) and is able to find most of the objects in the image (recall). Meanwhile, the last two graphs in the second row showing mAP50(B) and mAP50-95(B) also show a significant increase and stabilize at high values. A high mAP50 value indicates excellent detection performance at an IoU threshold of 0.5, while a similarly high mAP50-95 indicates consistent performance at various IoU threshold levels.

Overall, the results from these graphs illustrate a good training process. The YOLOv8 model can demonstrate good learning capabilities. It can be seen by the consistent loss reduction and significant improvement in evaluation metrics. It can also be seen that there is no sign of overfitting or underfitting which indicates that the model has reached a high level of performance in detecting garbage objects.

3.6. Model evaluation

After the YOLOv8 model training process has been completed, the next step is to evaluate the performance of the model using test data. This evaluation is important to assess the model's ability to detect objects in data that have never been seen before. In this process, we utilize the validation feature in YOLOv8 by using the results of the best training model, namely best.pt.

This evaluation is done by running the YOLOv8 model that has been trained on the test dataset that has been prepared. YOLOv8 will produce a visualization of the detection results in the form of a concussion matrix table. The main objective is to get a comprehensive picture of the model's ability to detect and classify various types of waste, as well as identify areas that may require further improvement. The visualization of the model evaluation results in the form of a confusion matrix table can be seen in Figure 8.

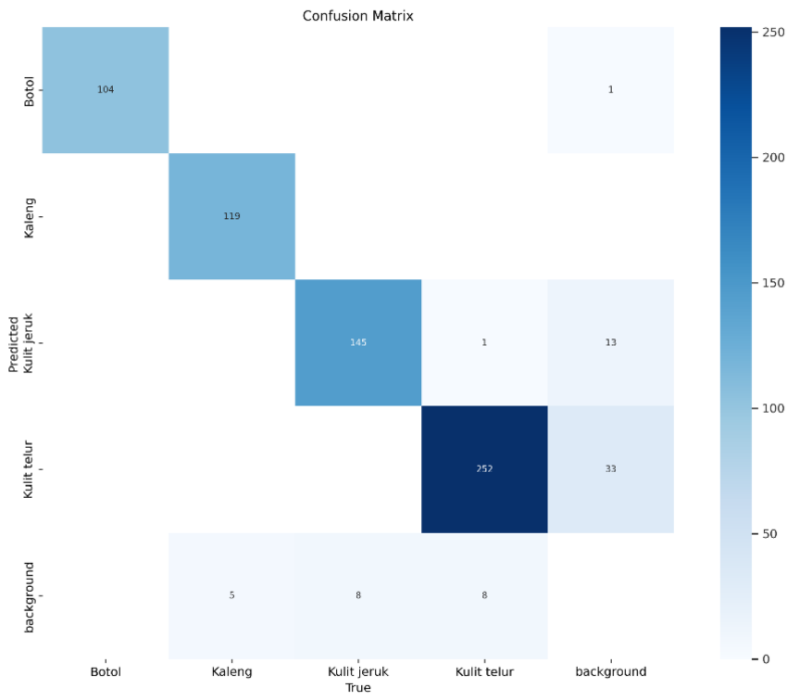

Fig. 8: Confusion Matrix Result

From the results of the confusion matrix above, it can be seen that there are 4 components in the confusion matrix, namely True Positive, False Positive, True Negative, and False Negative. The following is an explanation of these components in each object class in table 5.

Table 5: Confusion matrix details

Component / Object	True Positive	False Positive	True Negative	False Negative	Description
Plastic Bottle	104	0	663	1	• **TP** : 104 samples were correctly identified as bottles. • **FP** : No other objects were incorrectly identified as bottles. • **FN :** 1 Bottle sample not detected/misclassified • **TN** : 663 samples were correctly identified as bottles.
Drink cans	119	0	664	5	• **TP** : 119 samples were correctly identified as bottles. • **FP** : No other objects were incorrectly identified as bottles. • **FN :** Bottle sample not detected/misclassified • **TN** : 664 samples were correctly identified as bottles.
Orange peels	145	8	601	33	• **TP** : 145 samples were correctly identified as bottles. • **FP** : 8 non-orange peel objects misidentified as orange peels. • **FN :** 33 Bottle sample not detected/misclassified • **TN** : 601 samples were correctly identified as bottles.
Eggshells	252	8	475	33	• **TP** : 252 samples were correctly identified as bottles. • **FP** : 8 non-egg objects misidentified as eggshells. • **FN :** 33 Bottle sample not detected/misclassified • **TN** : 475 samples were correctly identified as bottles.

Based on the results of the confusion matrix data above, it can be concluded that the model performs very well in identifying bottles and cans with high accuracy and minimal errors. However, there are still challenges in detecting orange peels and eggshells where the model sometimes fails to recognize or misclassifies these objects. In addition, the model also has difficulty in distinguishing some backgorund objects which is characterized by the presence of false positives and false negatives in the backgorund category. Overall, the model has shown promising capabilities in detecting litter, with good performance for most classes. However, it cannot be denied that there is still a need for improvement in improving detection accuracy in the orange peel and eggshell classes, as well as distinguishing objects from the background.

After viewing and evaluating the model with the validation function in YOLOv8, the model obtained a Mean Average Precssion (Map) value of 0.968 which can be seen in the table below.

Table 6: mAP Results on Test Data

Images	Precision	Recall	mAP
140	0.938	0.960	0.968

Based on the test results table above, the YOLOv8 model gets an mAP value of 0.968 or 96.8% on the test data. This shows the average performance of the model in detecting and classifying objects on the test dataset. This mAP result indicates that the model has a good performance in detecting waste objects in the test data. The model shows a good balance between precision of 0.938 and recall of 0.960 which means that the model can detect most of the relevant objects with a fairly low error rate.

In addition to being evaluated by looking at the mAP value on the test data, the model is also evaluated by looking at the accuracy results. Where accuracy is used to measure the proportion of correct predictions of all predictions generated by the model. Accuracy is calculated based on the data in the confusion matrix in Figure 8. Calculation of accuracy results can be seen in the following example.

$$
\begin{aligned}
accuracy &= \frac{total\ correct\ predictions}{total\ all\ correct} \times 100\% \\
&= \frac{104 + 119 + 145 + 252}{104\ +\ 119\ +\ 145\ +\ 252\ +\ 1\ +\ 1\ +\ 13\ +\ 33\ +\ 5\ +\ 8\ +\ 8} \\
&= \frac{620}{689} \times 100\% \\
&= 89{,}98\%
\end{aligned}
\tag{1}
$$

It can be seen that the accuracy calculation result based on the data in the confusion matrix is 89.98%, this shows a pretty good performance. With this good value of mAP and accuracy results, it can be concluded that this YOLOv8l model has a fairly good performance in detecting garbage objects in the test data.

3.7. Integration System

After the model training process has been successful and the model that has been trained is then stored. Next is the stage of system integration with microcontrollers. In this integration process, Arduino UNO was chosen as the microcontroller module in the main processing unit. Arduino UNO was also chosen because of its ease of use and wide availability and has a fairly high compatibility with various sensors and actuators which causes ease of integration between components and also has low power consumption and a fairly portable size.

Fig. 9 Arduino uno module

The Arduino UNO plays an important role in processing the detection results from the model and determining the appropriate trash can to open. This determination decision is then translated into a control signal sent to the servo motor. After the data is received, the servo motor that serves as the actuator will move the trash can lid based on the signal received. In the integration process, this implementation uses 2 kinds of programming languages, namely python and c++. Python runs the detection model on real-time video input by the camera to classify garbage. Python also serves to create the interface of the detection window and is also responsible for processing the detection results and sending control signals to the Arduino UNO via serial communication.

3.8. System Testing

At this stage, testing was carried out in real-time at Taman Merdeka Binjai. The scenario in testing this system involves setting the distance between the camera position and the object. Testing was carried out in an open area and carried out with a focus on the 4 main objects of research, namely bottles, cans, eggshell orange peels. The distance tested in the test was set to less than 30cm between the camera and the object to allow detailed observation of the system's ability to detect objects at close range. The main focus of this test was to observe how well the detection and performance of the device in opening and closing the bin according to the detection results with the system integrated using a microcontroller. The evaluation includes the accuracy of object detection, indicated by the confidence value on the detection screen, as well as the mechanical response of the bin to the detection result. In this test, several inferences have been set in the detection program with the aim of optimizing overall system performance. Some of the inference settings in the program can be seen in table 7 below.

Table 7: Inference Setup Table

Parameter	Value	Description
Device	"0"	Using GPU Rendering.
Confidence Threshold	0,5 or 50%	Detected if confidence ≥ 0.5 or 50%..
Video Capture	1	Using external cameras.
Frame Size	Customize	Using the default resolution of the external camera in use.
Update Interval	10ms	Frame update frequency and detection.
Serial Communication	COM8,57600 baud	It uses port 8 with a data transfer rate of 57,600 bits/second.

After setting up the inference in the detection program, detection testing began with the following test results:

3.8.1. First Testing

Fig. 10 Bottle detection testing

The results of the first test in real-time by taking a screenshot of the detection program which is then displayed as shown in Figure 10. This test uses a bottle object as a detection sample, showing that the system is able to detect bottle objects with a confidence threshold of 0.94 and the mechanical response of the trash can that opens according to the type of garbage detected, namely inorganic.

3.8.2. Second Testing

Fig. 11 Drink can detection testing

The results in the second test are real-time by taking a screenshot of the detection program which is then displayed as shown in Figure 11. This test uses a can object as a detection sample, showing the system is able to detect a can object with a confidence threshold of 0.64 and the mechanical response of an open trash can in accordance with the type of garbage detected, namely inorganic.

3.8.3. Third Testing

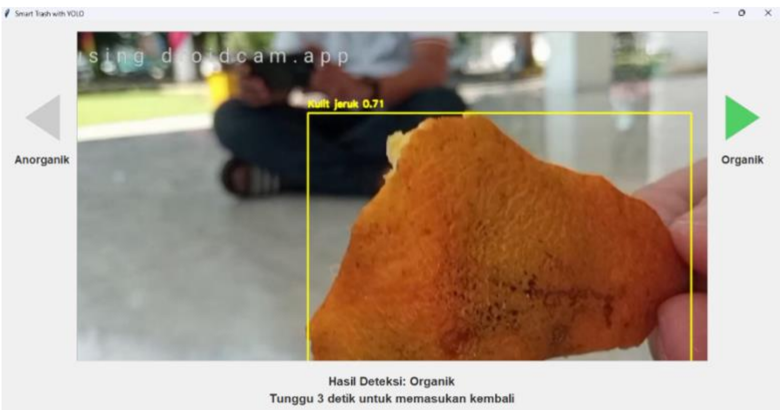

Fig. 12 Orange peel detection testing

The results in the third test are real-time by taking a screenshot of the detection program which is then displayed as shown in Figure 12. This test uses an orange peel object as a detection sample, showing that the system is able to detect an orange peel object with a confidence threshold of 0.71 and the mechanical response of the trash can that opens according to the type of waste detected, namely organic.

3.8.4. Fourth Testing

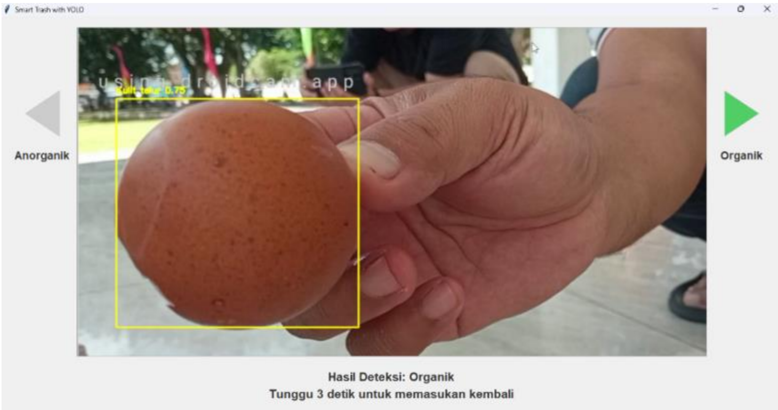

Fig. 13 Eggshell detection testing

The results in the fourth test are real-time by taking a screenshot of the detection program which is then displayed as shown in Figure 13. This test uses eggshell objects as detection samples, showing the system is able to detect eggshell objects with a confidence threshold of 0.75 and the mechanical response of the trash can that opens according to the type of waste detected, namely organic.

4. Conclusion

Based on tests from research that has been conducted using the YOLOv8l model in an automatic waste type detection system, this study successfully implements the YOLOv8l model to detect and classify types of waste, including bottles, cans, orange peels, and eggshells. This model shows high effectiveness with a Mean Average Precision value of 0.968, which indicates the model's ability to recognize and distinguish between types of waste in various conditions. Testing the system in real-time conditions at Taman Merdeka Binjai also shows consistent performance, with confidence threshold values that vary but remain above the 0.5 threshold. This shows that the system is reliable in detecting waste with good accuracy, despite variations in distance and type of waste detected.

Regarding the system's detection accuracy in identifying organic and inorganic waste types, the test results show that the system can effectively integrate the YOLOv8l-based visual detection component with the bin control mechanism using Arduino UNO and servo motors. The accurate mechanical response in opening the bin according to the detected waste type proves the effectiveness of the communication between the Python detection program and the Arduino microcontroller. Although the system performed well, the test results also indicated that the optimal performance of the system depends on the environmental conditions and the quality of the visual input, which opens up opportunities for further development in improving the system's adaptability to variations in lighting and camera device quality.

Acknowledgement

The authors would like to express their gratitude to everyone who provided valuable feedback and support during the completion of this work.

References

[1] Kementerian Lingkungan Hidup dan Kehutanan, “Data Timbulan Sampah Nasional,” Sistem Informasi Pengelolaan Sampah Nasional. Accessed: Mar. 31, 2024. [Online]. Available: https://sipsn.menlhk.go.id/sipsn/public/data/timbulan

[2] U. Sumarna *et al.*, “Promosi Kesehatan Mengenai Pengelolaan Sampah di Rw 13 Kelurahan Sukamentri Kecamatan Garut Kota Kabupaten Garut,” *Jurnal Kreativitas Pengabdian Kepada Masyarakat (PKM)*, vol. 6, no. 4, pp. 1475–1485, Apr. 2023, doi: 10.33024/jkpm.v6i4.9024.

[3] Kementerian Lingkungan Hidup dan Kehutanan Republik Indonesia, “Data Sampah Indonesia.”

[4] A. Gupta, N. Gupta, and J. Gupta Freelancer, “Machine Learning Based Prototype Working Model to Determine Category of Garbage Waste Using Raspberry Pi,” 2022, doi: 10.21203/rs.3.rs-2056856/v1.

[5] M. Panmuang and C. Rodmorn, “Garbage Detection using YOLO Algorithm for Urban Management in Bangkok,” *WSEAS Transactions on Computer Research*, vol. 12, pp. 236–243, 2024, doi: 10.37394/232018.2024.12.23.

[6] Z. Ahmed and S. Sadat Shanto, “Performance Analysis of YOLO Architectures for Surgical Waste Detection in Post-COVID-19 Medical Waste Management,” *Malaysian Journal of Science and Advanced Technology*, pp. 1–9, Jan. 2024, doi: 10.56532/mjsat.v4i1.232.

[7] Z. Lun, Y. Pan, S. Wang, Z. Abbas, M. S. Islam, and S. Yin, “Skip-YOLO: Domestic Garbage Detection Using Deep Learning Method in Complex Multi-scenes,” *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, Dec. 2023, doi: 10.1007/s44196-023-00314-6.