Automatic Buyer Machine for Beverage Waste

Barry Linando, Muhammad Jembar Jomantara, Wiedjaja Atmadja*

Computer Engineering Department, Faculty of Engineering, Bina Nusantara University,
Palmerah Jakarta Barat 11480, Indonesia
*Corresponding author: steff@binus.edu

Abstract. Automatic Buyer Machine for Beverage Waste is an automatic machine for buying beverage packaging waste using a Raspberry Pi as the main controller and the Convolutional Neural Network (CNN) method. By using object detection, the machine can recognize the classification of trained objects. If the object detection is successful in classifying the object, then the object will be accepted by the machine and then directed to the container according to each classification. This experiment uses the Pre-Trained SSDLITE_MOBILENET_V2_COCO model and works well on a Raspberry Pi. The results of the experiment yield an accuracy rate of 95% when the detection model is trained using an augmentation configuration. It was concluded that the system is able to detect beverage packaging objects according to the classification.

Keywords: Object Detection, Raspberry Pi, Reverse Vending Machine

I. INTRODUCTION

According to the Law of the Republic of Indonesia Number 18 of 2008 on Waste Management, waste is the residue of daily human activities and/or natural processes in solid form. Garbage is solid waste material from household, markets, offices, lodging houses, hotels, restaurants, and industry activities. Another definition of Garbage is a by-product of human activities that have been used [1]. Every human activity produces waste or garbage, and the waste volume is proportional to the level of daily consumption of goods or materials [2].

The type of waste itself is divided into two parts, namely organic waste, and inorganic waste. Organic waste is a type of waste that is easily decomposed, contrary to inorganic waste, thus decomposition of organic waste will occur faster than inorganic waste. An example of inorganic waste is beverage packaging waste, whether made from plastic, cans, or glass. According to the Indonesian Central Bureau of Statistics (Badan Pusat Statistik - BPS), an increase in plastic waste every year in various regions caused Indonesia to be the second largest waste-producing country with 187.2 tons of waste in 2015 [3]. However, most of the use of bottles or beverage packaging in Indonesia itself is made of plastic where we know that plastic is one of the materials that is very difficult to decompose by the soil, which takes about 450 years. The use of plastic in human life is increasing. This increase occurs because plastic is light, practical, economical and can replace the function of other goods [4]. In the global industry, the production of synthetic plastics is one of the fastest growing fields. Despite the fact that plastics have been used in everyday life for 100 years, the beginning of large-scale production was in the 1950s [5].

However, there are several waste treatment techniques that have been applied in several places. One of them is point-of-sale recycling which refers to the practice of returning glass bottles for a refund [6]. Several researchers have thought of a solution to this waste problem, one of which is by making a machine called a Reverse Vending Machine that applies the cash-from-trash concept where recycled plastic can be used to produce useful products [7]. A similar machine was also designed by researchers in Malaysia which uses an Arduino microcontroller and several sensors to detect metal waste inserted by user who will then get a reward [8]. A similar system was also created by researchers from India using a Raspberry Pi and applying an object detection model, namely the Haar Cascade Classifier [9].

Based on the problems and previous research, we propose to design a tool called Automatic Buyer Machine for Beverage Waste. This machine will return money to users who insert inorganic waste in the form of beverage containers made of plastic, cans, or glass. With these monetary incentives, it is desirable that there will be an increase in public awareness and enthusiasm for environmental cleanliness and support of the
inorganic waste recycling culture. This machine is designed using Raspberry Pi 4B as its main computing component by applying object detection model as a method to distinguish the types of packaging waste, such as plastic, cans, and glass. To capture object images, this machine is equipped with a USB WebCam. Other supporting system used is a NEMA17 Stepper Motor with a motor driver (TB6560) as a conveyor driver to deliver objects into or out of the machine. Also, this machine is equipped with a laser which is combined with an LDR sensor to detect any incoming object. Then, to direct objects according to their classification result, this machine is equipped with a pair of SG90 Servo Motors.

II. RESEARCH METHOD

In this section, the configuration of object detection model training and testing will be explained, followed by the hardware design.

A. Configuration of Object Detection Model Training

The Object Detection model used in detecting beverage packaging objects is a custom model that is trained by utilizing the Pre-Trained Detection Models provided by TensorFlow1. In training this model, datasets are needed in the form of pictures that have been labelled according to their respective classifications. In this experiment, we used several types of beverage packaging, namely as follows:

![Types of beverage packaging that used in the detection model training process](image)

*Figure 1. Types of beverage packaging that used in the detection model training process*

It appears that the sample photos of each type of beverage packaging were taken in different quantities where 10 pictures taken for each plastic beverage packaging, 20 pictures taken for each can, and 30 pictures taken for each glass. This is due to the objective to conduct experiments and observations on the effect of various patterns of beverage packaging on a class. For example, for packaging made from cans, 100 pictures were obtained with 5 different patterns (shapes to various colours), which were then classified as “Cans”. Meanwhile, for the glass packaging, 60 pictures were obtained with 2 different patterns (same shape, but different colours), which were then classified as “Glass”.
The sample pictures were taken from the same camera and the position of the object was at the same detection position (according to the position of the LDR and Laser Sensors). The following sample picture was taken from a camera 45 cm above the conveyor.

![Camera position](image)

**Figure 2.** Camera position

The pictures produced from this camera have a size of 1024x600 pixels with file size of around 190 KB to 210 KB. The training process is carried out by utilizing Google Colabs, where every user gets a maximum of 12 hours of GPU borrowing time for free. When the training process is complete, the detection model is tested on a Raspberry Pi 4B using TensorFlow Lite. The decision to use TensorFlow Lite is due to its capability to produce higher FPS with less CPU requirements compared to TensorFlow.

**B. Configuration of Object Detection Model Testing**

In this research process, the test conditions were the same as the conditions when the dataset image was taken (same camera and conveyor positions) and each type of beverage packaging will be tested 20 times to get more accurate result. Therefore, it is expected that the attained IoU is greater than the typical 0.95 in order to get the “Perfect” annotation [10]. Thus, this test has a failure criterion of:

- Wrong Classification, if the classification of detection results does not match the classification in the Training process.
- IoU Scores < 0.95 typical, if the resulted IoU Score is less than 95% typical.

The experiment was carried out 4 times, as following:

1. Experiment 1: detection model using SSDLITE_MOBILENET_V2_COCO with 5 classes
2. Experiment 2: detection model using SSDLITE_MOBILENET_V2_COCO with 3 classes
3. Experiment 3: detection model using SSD_MOBILENET_V1_FPN_COCO with 3 classes
4. Experiment 4: detection model using SSDLITE_MOBILENET_V2_COCO with 3 classes and Augmentation Configuration

When experiment 1 and 2 were compared, the resulted data is on the effect of number of classes. When experiments 2 and 3 are compared, the result is the data regarding the effect of the pretrained detection model used. Lastly, when experiments 2 and 4 are compared, data regarding the effect of augmentation configuration was attained.

In this experiment, observations were also made regarding the success of the system in terms of directing detected objects to their respective containers. The directing of detected objects is carried out by a pair of SG90 servo motors equipped with pointing sticks as shown in **Figure 3**.
In addition, observations were also done on the success of the system in terms of displaying detected objects on the screen. The number of objects detected, and the accumulated money earned are displayed so that the user can know the status. The screen display is depicted in following image:

![Image of screen display]

**Figure 4. User Interface**

### C. Hardware Design

Based on the Block Diagram, most of the components are directly connected to the Raspberry Pi 4B such as LDR Sensor, SG90 Servo Motor, USB webcam camera, and a 7-inch LCD. Raspberry Pi 4B is used as the core system where the input and output data processing will occur in it. The Adaptor 5VDC 3A is used as a power source for the Raspberry Pi 4B where the current required is 3A. Camera USB Webcam is used as a tool to capture images of every incoming object on the conveyor. Laser and LDR sensors are used as object detectors. When the sensor detects an object (the laser light is blocked by an object), the system will send an input data to the Raspberry Pi 4B, and then the conveyor will stop. Meanwhile, three other LDR sensors are
placed in each container to determine whether the container is full or not. The 7 Inch LCD screen is used as a tool to display information about the detected object and display the accumulated money earned by the user. The NEMA17 Stepper Motor is used to move the conveyor inward and outward. The SG90 Servo Motors are used to direct the detected objects to their respective containers according to their classification. The 12VDC 5A adapter is used as a power source for the TB6560 Stepper Motor Driver, Laser and LED Strip.

![Figure 6. Schematic of the System](image)

In this system, a comparator IC, LM339, is used where this component functions to convert the analog signal from the LDR sensor into a digital signal which is then sent to the Raspberry Pi 4B. The LM339 is a Quad-Comparator Op-Amp which means that there are 4 comparator systems. All the output pins of the LM339 are then connected to the Raspberry Pi 4B as digital inputs.

This system uses two SG90 Servo Motors, each of which requires 1 output pin from the Raspberry Pi 4B. This system also uses the TB6560 Stepper Motor Driver which requires 2 output pins namely DIR and STEP from the Raspberry Pi 4B. In addition, the system also added a port for the Raspberry Pi 4B Cooling Fan and a current splitting system (parallel circuit) for 4 Lasers.

**III. RESULT AND DISCUSSION**

**A. Experiment 1: detection model using SSDLITE_MOBILENET_V2_COCO with 5 classes**

The training was carried out using a dataset consisting 310 images of several types of beverage packaging, as shown in Figure 1. In this experiment, the sample images will be divided into 5 different classes, namely:

| Class        | Description                                           |
|--------------|-------------------------------------------------------|
| Plastic_Large| consisting of beverage packaging made of plastic with a capacity of 1.5 Liters |
| Plastic_Medium| consisting of beverage packaging made of plastic with a capacity of 500ml to 600ml |
| Plastic_Small| consisting of beverage packaging made of plastic with a capacity of 200ml to 350ml |
| Cans         | consisting of beverage packaging made from cans with all sizes |
| Glass        | consisting of beverage packaging made of glass with all sizes |

Each image is given a label on the position of the object with the label name according to their respective classification. Once labelled, the sample images were then trained (Training Process) using the SSDLITE_MOBILENET_V2_COCO Pre-Trained Detection Model with a configuration of Batch Size = 10 and not using Augmentation.
As depicted in Figure 7, the training was carried out up to around 130 thousand steps and it appears that the loss was reduced quite significantly starting from the first up to the 30 thousand steps. The entire training process is carried out for approximately 12 hours at a speed of 0.3s/step. After the model is trained, the model is exported into a TFLite model where the output obtained is in the form of .pb file format which will be then converted into a .tflite file format so that it can be used on the Raspberry Pi for further model testing. When testing on the tool was carried out, around 50-60% of the CPU on the Raspberry Pi 4B was used with FPS obtained around 3.4 to 4.2. Table 1 shows the test results of each type of beverage packaging that was each tested 20 times.

**Table 1. Experiment Result using Model SSDLITE_MOBILENET_V2_COCO with 5 classes**

| Sample Name             | Success | Failure | IoU Scores |
|-------------------------|---------|---------|------------|
|                         |         | Wrong Classification | < 0.95 typical |
| Club 1.5L               | 12      | 8       | 0          |
| Club 1.5L (No Label)    | 12      | 8       | 0          |
| AQUA 1.5L               | 18      | 2       | 0          |
| AQUA 1.5L (No Label)    | 17      | 3       | 0          |
| Le Minerale 1.5L        | 15      | 5       | 0          |
| Club 600ml              | 18      | 2       | 0          |
| AQUA 600ml              | 15      | 5       | 0          |
| Le Minerale 600ml       | 20      | 0       | 0          |
| YouC1000 600ml          | 17      | 3       | 0          |
| Pocari 500ml            | 16      | 4       | 0          |
| Pocari 350ml            | 20      | 0       | 0          |
| Chocolatos 200ml        | 19      | 1       | 0          |
| AQUA 330ml              | 17      | 3       | 0          |
| Kapal Api White Coffee 200ml | 20 | 0  | 0         |
| Adem Sari Ching Ku 350ml| 19      | 1       | 0          |
| Adem Sari Ching Ku 320ml| 20      | 0       | 0          |
| AW Sarsaparila 330ml    | 20      | 0       | 0          |
| Hemaviton C1000 330ml   | 19      | 0       | 1          |
| Milo Calsium 240ml      | 20      | 0       | 0          |
From the results shown in Table 1, it can be concluded that the model still cannot appropriately detect objects, as there are 26 failures caused by wrong classification. The most inaccurate occurrence is when detecting Club 1.5L bottles with labels and without packaging labels where the classification should be Plastic_Large, but the results obtained are 8 times failure due to misclassification (detected as Plastic_Medium). From all the 440 tests carried out, 392 successes were obtained, and 48 failures were obtained, hence the accuracy value attained is 89.091%. In this experiment, a 100% success in directing objects to their respective containers was attained, and a 100% success in displaying detected objects on the screen was also attained.

B. Experiment 2: detection model using SSDLITE_MOBILENET_V2_COCO with 3 classes

The training was carried out using a dataset of 310 images of several types of beverage packaging, as shown in Figure 3. In this experiment, the sample images will be divided into 3 different classes, namely:

- **Plastic** consisting of beverage packaging made of plastic with all sizes
- **Cans** consisting of beverage packaging made from cans with all sizes
- **Glass** consisting of beverage packaging made of glass with all sizes

Each picture is given a label on the position of the object with the label name according to their respective classification. Once labelled, the sample images were then trained (Training Process) using the SSDLITE_MOBILENET_V2_COCO Pre-Trained Detection Model with a configuration of Batch Size = 10 and not using Augmentation.

![Figure 8. Loss Graph from TensorBoard based on SSDLITE_MOBILENET_V2_COCO Training Results Data with 3 classes](image)

As shown in Figure 8, the training was carried out up to around 154 thousand steps and it appears that the loss was reduced quite significantly from the first step up to 30 thousand steps. The entire training process is carried out for approximately 15 hours at a speed of 0.3s/step. After this model is trained, it is then exported and converted into a TFLite model, which then tested in Raspberry Pi.

When testing on the tool was carried out, around 50-60% of the CPU on the Raspberry Pi 4B was used with FPS obtained around 3.4 to 4.2. Table 2 shows the test results of each type of beverage packaging that was each tested 20 times.

| Sample Name          | Success | Failure | Wrong Classification | IoU Scores < 0.95 typical |
|----------------------|---------|---------|----------------------|---------------------------|
| Club 1.5L            | 11      | 0       | 0                    | 9                         |
| Club 1.5L (No Label) | 14      | 0       | 0                    | 6                         |
| Product                  | Success | Failures | Total |
|-------------------------|---------|----------|-------|
| AQUA 1.5L               | 18      | 0        | 2     |
| AQUA 1.5L (No Label)    | 18      | 0        | 2     |
| Le Mineral 1.5L         | 15      | 0        | 5     |
| Club 600ml              | 18      | 0        | 2     |
| AQUA 600ml              | 16      | 0        | 4     |
| Le Mineral 600ml        | 20      | 0        | 0     |
| YouC1000 600ml          | 16      | 0        | 4     |
| Pocari 500ml            | 16      | 0        | 4     |
| Pocari 350ml            | 20      | 0        | 0     |
| Chocolatos 200ml        | 19      | 0        | 1     |
| AQUA 330ml              | 18      | 0        | 2     |
| Kapal Api White Coffee 200ml | 20      | 0        | 0     |
| Adem Sari Ching Ku 350ml | 20      | 0        | 0     |
| Adem Sari Ching Ku 320ml | 20      | 0        | 0     |
| AW Sarsaparila 330ml    | 20      | 0        | 0     |
| Hemaviton C1000 330ml   | 19      | 0        | 1     |
| Milo Calsium 240ml      | 20      | 0        | 0     |
| Nescafe Latte 240ml     | 19      | 0        | 1     |
| YouC1000 Jeruk 140ml    | 20      | 0        | 0     |
| YouC1000 Apel 140ml     | 20      | 0        | 0     |
| **TOTAL**               | 397     | 43       |       |

From the data shown in Table 2, it can be concluded that the model still cannot appropriately detect objects. As there are 43 failures due to IoU Scores under the 0.95 typical. The most inaccurate occurrence is when detecting a Club 1.5L bottle where it failed 9 times and it was detected as Plastic, with IoU Scores of 0.55 to 0.70. From all the 440 tests carried out, 397 was a success and 43 was a failure, hence the accuracy value attained is **90.227%**. Through the experiment, 100% success in directing objects to their respective containers was attained, and 100% success in displaying detected objects on the screen was also attained.

C. Experiment 3: detection model using SSD_MOBILENET_V1_FPN_COCO with 3 classes

The training was carried out using a dataset consisting of 310 images of several types of beverage packaging as shown in Figure 3. In this experiment, the sample images will be divided into 3 different classes, namely:

- Plastic, consisting of beverage packaging made of plastic with all sizes
- Cans, consisting of beverage packaging made from cans with all sizes
- Glass, consisting of beverage packaging made of glass with all sizes

Each picture is given a label on the position of the object with the label name according to their respective classification. Once labelled, the sample images were then trained (Training Process) using the SSD_MOBILENET_V1_FPN_COCO Pre-Trained Detection Model with a configuration of Batch Size = 10 and not using Augmentation.
From the graph shown in Figure 9, the training was carried out up to around 15 thousand steps and it shows that the loss was reduced quite significantly from the first up to the 4 thousand steps. The entire training process is carried out for approximately 24 hours at a speed of 0.8s/step.

Although the training results are very good because the loss reaches about 0.01, the test results show the opposite (very bad) because the CPU usage on the Raspberry Pi 4B is around 90-99% with the FPS obtained around 0.2 so it cannot be used, and data retrieval can be done.

D. Experiment 4: detection model using SSDLITE_MOBILNET_V2_COCO with 3 classes and Augmentation Configuration

The training was carried out using a dataset consisting of 310 images of several types of beverage packaging as shown in Figure 3. In this experiment, the sample images will be divided into 3 different classes, namely:

- Plastic consisting of beverage packaging made of plastic with all sizes
- Cans consisting of beverage packaging made from cans with all sizes
- Glass consisting of beverage packaging made of glass with all sizes

Each picture is given a label on the position of the object with the label name according to their respective classification. Once labelled, the sample images were then trained (Training Process) using the SSDLITE_MOBILNET_V2_COCO Pre-Trained Detection Model with a configuration of Batch Size = 10 and using Augmentation, which is as follows:

- random_vertical_flip,
- random_rgb_to_gray,
- random_adjust_brightness,
- ssd_random_crop

Figure 10. Loss Graph from TensorBoard based on SSDLITE_MOBILNET_V2_COCO Training Results Data with 3 classes and Augmentation Configuration
As shown from the graph in Figure 10, the training was carried out up to around 100 thousand steps and it shows that the loss was reduced quite significantly from the first up to the 30 thousand steps. The entire training process was carried out for approximately 20 hours at a speed of 0.7s/step.

When testing on the tool was performed, the CPU usage on the Raspberry Pi 4B was around 65% with FPS obtained around 3.2 to 3.9. Shown in Table 3 are the test results of each type of beverage packaging which was each tested 20 times.

| Sample Name | Success | Failure | Wrong Classification | IoU Scores < 0.95 typical |
|-------------|---------|---------|---------------------|-------------------------|
| Club 1.5L   | 17      | 0       | 3                   |                         |
| Club 1.5L (No Label) | 18 | 0 | 2 |                         |
| AQUA 1.5L   | 19      | 0       | 1                   |                         |
| AQUA 1.5L (No Label) | 18 | 0 | 2 |                         |
| Le Minerale 1.5L | 19 | 0 | 1 |                         |
| Club 600ml  | 18      | 0       | 2                   |                         |
| AQUA 600ml  | 18      | 0       | 2                   |                         |
| Le Minerale 600ml | 19 | 0 | 1 |                         |
| YouC1000 600ml | 18 | 0 | 2 |                         |
| Pocari 500ml | 20 | 0 | 0 |                         |
| Pocari 350ml | 20 | 0 | 0 |                         |
| Chocolatos 200ml | 19 | 0 | 1 |                         |
| AQUA 330ml  | 18      | 0       | 2                   |                         |
| Kapal Api White Coffee 200ml | 20 | 0 | 0 |                         |
| Adem Sari Ching Ku 350ml | 19 | 0 | 1 |                         |
| Adem Sari Ching Ku 320ml | 20 | 0 | 0 |                         |
| AW Sarsaparila 330ml | 19 | 0 | 1 |                         |
| Hemaviton C1000 330ml | 19 | 0 | 1 |                         |
| Milo Calisum 240ml | 20 | 0 | 0 |                         |
| Nescafe Latte 240ml | 20 | 0 | 0 |                         |
| YouC1000 Jeruk 140ml | 20 | 0 | 0 |                         |
| YouC1000 Apel 140ml | 20 | 0 | 0 |                         |
| TOTAL       | 418     | 22      |                     |                         |

From the data shown in Table 3, it can be concluded that the detection model can appropriately detect objects. There were 22 failures caused by IoU Scores < 0.95 typical, where the main cause is the difference in the pattern of the object when the object is rotated little by little, causing the IoU Score to drop to around 0.80 to 0.85. From all the 440 tests performed, 418 times were a success, and 22 times were a failure, hence the accuracy value of 95% was obtained. In this experiment, 100% success in directing objects to their respective containers and displaying detected objects on the screen were attained.

IV. CONCLUSION

- Experiment 1 Detection Model using SSDLITE_MOBILENET_V2_COCO with 5 Classes produces an accuracy rate of 89.091%.
- Experiment 2 regarding the Detection Model using SSDLITE_MOBILENET_V2_COCO with 3 Classes produces an accuracy rate of 90.227%.
- Experiment 4 regarding the Detection Model using SSDLITE_MOBILENET_V2_COCO with 3 Classes and Augmentation Configuration produces an accuracy rate of 95%.
- The number of classes does not affect the detection accuracy rate.
- Augmentation configuration has a significant effect on the detection accuracy rate.
• The SSD_MOBILENET_V1_FPN_COCO model can be used on the Raspberry Pi 4B but produces a very small FPS and uses a very large CPU.
• The variety of beverage packaging patterns in a class affects the accuracy of detection. It is evident from the results of Experiment 1, 2, and 4, that glass packaging which only has 2 different varieties (same shape, but different colors) produce a 100% success rate (20 trials each). Meanwhile, plastic packaging that has many varieties (different sizes, colors, and packaging patterns) produce a much lower success rate.
• Based on the results of Experiments 1, 2 and 4, a 100% success rate was obtained in directing objects to their respective Containers.
• Based on the results of Experiments 1, 2 and 4, a 100% success rate was obtained in displaying data on the screen.

REFERENCE
[1] Sucipto C D 2012 Teknologi Pengolahan Daur Ulang Sampah (Yogyakarta: Gosyen Publishing)
[2] Sejati K 2009 Pengolahan Sampah Terpadu dengan Sistem Node, Sub Point, Center Point (Padang: Kanisius)
[3] Baldan S K, Aditiya A, Umiati V F, Yudhiana, T, Hafifah D N and Indreswari R 2018 Gerakan Vertigo Hangingplant (Ver-Hang) Pemanfaatan Sampah Botol Plastik di Dukuh Selo Tawangsari Sukoharjo Prosiding Seminar Nasional dan Internasional Fakultas Pertanian Universitas Muhammadiyah Semarang, 2 57-62
[4] Asroni M, Djiwo S and Setyawan E Y 2018 Pengaruh Model Pisau Pada Mesin Sampah Botol Plastik Aplikasi dan Inovasi IPTEK’S “SOLIDARITAS” I 29-33
[5] Geyer R, Jambeck J R and Law K L 2017 Production, use, and fate of all plastics ever made. Science Advances 3 7
[6] Kennedy S M, Copes R, Bartlett K H and Brauer M 2004 Point-of-Sale Glass Bottle Recycling: Indoor Airborne Exposures and Symptoms Among Employees Occupational and Environmental Medicine 61 628-35
[7] Shambi S and Dahiya P 2020 Reverse Vending Machine for Managing Plastic Waste Int. J. of System Assurance Engineering and Management 11 635-40
[8] Wong K K, Samah N A A, Sahimi M S and Othman W 2019 Development of Reverse Vending Machine using Recycled Materials and Arduino Microcontroller Int. J. of Engineering Creativity and Innovation 17-16
[9] Yaddanapudi S D, Makala B P, Yarlagadda A, Sapram C T, Parsa S T and Nallamadugu S 2019 Collection of Plastic Bottles by Reverse Vending Machine using Object Detection Technique Materials Today Proc.
[10] Hofesmann E 2020 IoU a better detection evaluation metric (Toward Data Science. Retrieved from https://towardsdatascience.com/oiu-a-better-detection-evaluation-metric-45a511185be1)