A Smart Trash Bucket with Sensor-Based Monitoring Capability for Clean Environments

Hyeonah Jung, Minjae Kim, Jiseop Lee and Jaehoon (Paul) Jeong
Department of Computer Science and Engineering
Sungkyunkwan University, Suwon, Republic of Korea
Email: {hyeonah214, lovely01ty, odo2718, pauljeong}@skku.edu

***Abstract*—Over the recent years, the growth of Internet of Things (IoT) technology resulted in the emergence of numerous smart devices. The smart devices using IoT technology, are often integrated with Artificial Intelligence (AI), Robotics, or different intelligence mechanisms. In this paper, we develop a sensor-based smart trash bucket, leveraging the concepts of AI. The primary objective of this paper is to create an automatic trash bucket which can inform its user of its loading status of the inside of the bucket through a ultrasonic sensor and can automatically detect the level of environmental cleanness around it through a camera-based object detection algorithm.. All of these functions operate in real time, allowing users to keep track of the status of both the inside and outside of the bucket via a website.**

***Index Terms*—Trash bucket, IoT, AI, detection, monitoring, clean environment**

I. INTRODUCTION

People nowadays are moving to cities with tall apartments for better living and facilities. The smart devices are likely to make our living comfortable. Since tall buildings are popular, the cleaning and maintenance of these building is a major challenge. Naturally, this is increasing the workload of the cleaning staff. It becomes quite tough to move up and down buildings on a regular basis for managing a large number of trash cans. This motivates us to develop a smart trash bucket management system that uses a combination of Internet of Things (IoT) and Artificial Intelligence (AI) technologies to track the condition of trash cans in real time.

The key idea of our proposed model is that the functions are compactly adjusted by setting specific users. Cleaning workers are set as users and system functions are selected according to their standards. The functions are divided into the following two categories: The first is to measure, indicate the amount of trash in the trash bucket. The second is to classify and detect trash objects placed around the trash bucket. Based on previous research work, we aim to design and implement a smart trash bucket to perform these two functions. The function of detection and classification is based on the combination of IoT and AI model. Through the YOLOv4 model [1], [2], we classify garbage into twelve types, learn it using AI, and classify it through images in real time.

In this paper, a smart trash bucket management system was proposed on the basis of the combination of the IoT and AI technologies. Once an AI model for a smart trash bucket is developed, cleaners will be able to work more efficiently and with less manual labor. In addition to helping cleaning workers, the system is expected to help users of trash buckets in the future. The YOLOv4 [1] model used in this paper, not only performs object detection but also performs object classification, which is expected to make it more efficient to perform garbage-related operations such as garbage separation and collection.

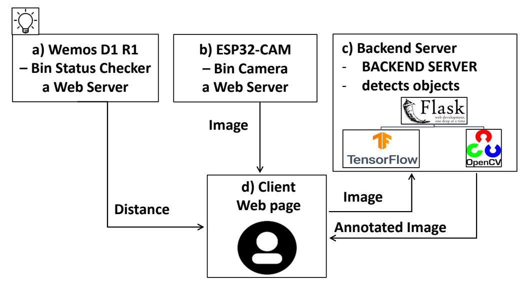

Fig. 1. Design of Smart Trash Bucket

II. DESIGN AND IMPLEMENTATION

In this section, we describe, the design and implementation of smart trash bucket. Recall that a smart trash bucket aims to enable real-time monitoring of both the inside and outside of the bucket so that the monitoring results can be displayed and managed on the website. The design for our system consists of four parts: a) Wemos D1 R1, b) ESP32-CAM, c) backend server, and d) client web page. Note that Fig. 1 illustrates the design of Smart Trash Bucket.

a) ***Wemos D1 R1:*** For the Wemos board, an additional library called "ESP Async Web Server" for ESP8266 is used to let the board work as a web server. When a web page is requested, the distance measurement and calculation are operated, and the distance is returned as a short text file (MIME type: "text/plain"). The original policy needs to be declared manually as 'any origin' for the HTTP response header, so that the client web page is assured to fetch the information from the server. Additionally, the distance is measured and at every one second independent of the web request, the board can turn the LED ON or OFF depending on the distance. The LED is turned ON if the distance is calculated to be less than 100 mm or more than 10,000 mm. The latter case is observed to happen if the sensor is blocked completely by something.

*b) **ESP32-CAM**:* For ESP32-CAM, the example web server program provided with ESP32 libraries for Arduino IDE is used for the board to work as a web server. It features a web page with various camera configurations, like resolutions, brightness, or saturation. There are two endpoints, “capture” and “stream” which provide a still JPEG image and motion JPEG video stream, respectively.

*c) **Backend Server**:* The backend server is based on Flask, a web framework written in Python. In November 2021, an anonymous person with the pseudonym Qone2 made a Flask-based object detection web server with YOLOv4 [1] weights, TensorFlow, and OpenCV-python [2]. To make the server work, YOLOv4 weights need to be converted into a TensorFlow model. The server provides four APIs: “by-image-files”, “by-image-file”, “by-url-list”, and “by-url”. All of these results are detected by TensorFlow with OpenCV-python. In our system, only “by-image-file” API is needed by our design. Additionally, in the original server, the server only stores images with detected results annotated temporarily and sends them as a response, but we modified it to make the server store annotated images as static files and send URIs as a response.

*d) **Client Web Page**:* For the web page, there is a premise that a network manager needs to know which system has which IP addresses for the distance checker and the camera. With this premise, the IP addresses need to be configured into the Script part of the web page. The web page uses the fetch web API, which can fetch a web resource across the network. With the fetch API, the page gets the distance information from Wemos D1 R1, and the image around the trash bucket from ESP32-CAM. The distance is directly processed in the web page since manipulating a single small number is not a big deal, while the image needs to be sent to the backend server. To send images to the server, the fetch API is used again. As explained in the backend part, the response for the request with an image is the URI for the annotated image. After the response, the gathered information including the distance and the annotated image is shown on the web page. The above procedure is set to occur every one second, which will be optimal for end users.

III. Performance Evaluation

We took two tests with a client web page: (i) a distance test with a real trash bucket in the middle of the development and (ii) a detection test in an indoor environment with various wastes at the end. As shown in Fig. 2, the distance test was successful, as the LED is ON when the bucket is full and is OFF when it is not full. The web page also shows the distances adequately. The detection test, however, was imperfect. The web page can show the annotated images well, but the detection was not always successful. For the detection, before the final detection experiment with a client, we had taken another experiment with YOLOv4 on Darknet, only to test the detection. As shown in Fig. 3, it detects well if there is a single object or not. However, sometimes mis-classification occurs, and more than two objects cannot usually be detected at the same time.

Fig. 2. Distance test. Above: a bucket not filled; Below: a full bucket.

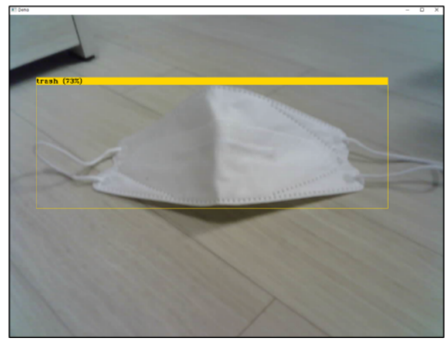

Fig. 3. Detection of a face protective mask

IV. Conclusion

With the development of IoT technology, smart trash buckets have emerged as one of the applications. Existing smart trash buckets cannot be used because they were not sufficiently managed due to the lack of object detection and classification. In this paper, we developed a well-managed smart trash bucket, using both IoT and AI technologies. It helps the users to classify garbage using AI and check the amount of garbage in trash buckets using IoT sensors. This will be easy for the cleaners to manage trash buckets. As future work, we will enhance the capability of object detection and classification for multiple objects through the localization of trash objects.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2020R1F1A1048263) and the Institute for Information & Communications Technology Planning & Evaluation (IITP) grant (No. 2019-0-01343, Regional strategic industry convergence security core talent training business). We authors sincerely appreciate the proofreading and editing of Eshita Rastogi. Note that Jaehoon (Paul) Jeong is the corresponding author.

References

[1] YOLO, “YOLOv4$_and_yolov4.weight_pre_trained dataset,''$ 2020, $https://github.com/AlexeyAB/darknet.$

[2] YOLO4, “Qone2. YOLOv4 Object Detection API Server,” 2021, https://github.com/Qone2/YOLOv4-Object-Detection-API-Server.