

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

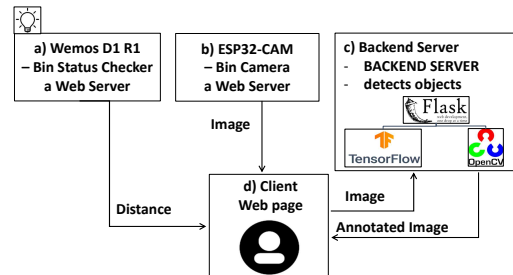


Fig. 1. Design of Smart Trash Bucket

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.

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.

