Boston University
Electrical & Computer Engineering
EC463 Senior Design Project

First Prototype Test Report

Ecobin

by

Team 9
Ecobin

Team Members

Aditya Wikara adwikara@bu.edu
Hayato Nakamura hayaton@bu.edu
Kevin Sinaga kfsinaga@bu.edu
Shreenidhi Jayaram shreej@bu.edu
Charles Thao cthao19@bu.edu

Ecobin

Table of Contents

1. Objective:

For the first prototype testing, Ecobin successfully completed its initial stage in integrating its hardware and image recognition software. The main objective of the prototype is to be able to detect simple objects, such as plastic bottles, and classify it as recyclable or trash. The team designed and implemented an embedded system with Raspberry Pi (henceforth RasPi or Pi). This prototype also explored concepts of cloud computing by simulating a cloud environment.

2. Equipment & Setup:

The setup is divided into two parts: hardware and software.

1. Hardware components:
    - Raspberry Pi 3 B+ (with 32GB SanDisk SDHC Class 10 card)
    - Raspberry Pi Camera Module v2
    - PIR Motion Sensor
    - LED Strips (5050 SMD RGBW,12V, 0.9W)
    - LEDs (green, red, white)
    - 12W Power Adapter (Pi3)
    - Power Supply Unit (to power LED Strips)
    - Transistor (npn-BJT PN2222A)

2. Software components:

    Python 3 scripts:

- Capture photos in JPEG format
- Object recognition
    - Tensorflow
    - Keras
    - Numpy, Scipy
- Signal handling
    - Raspi GPIO

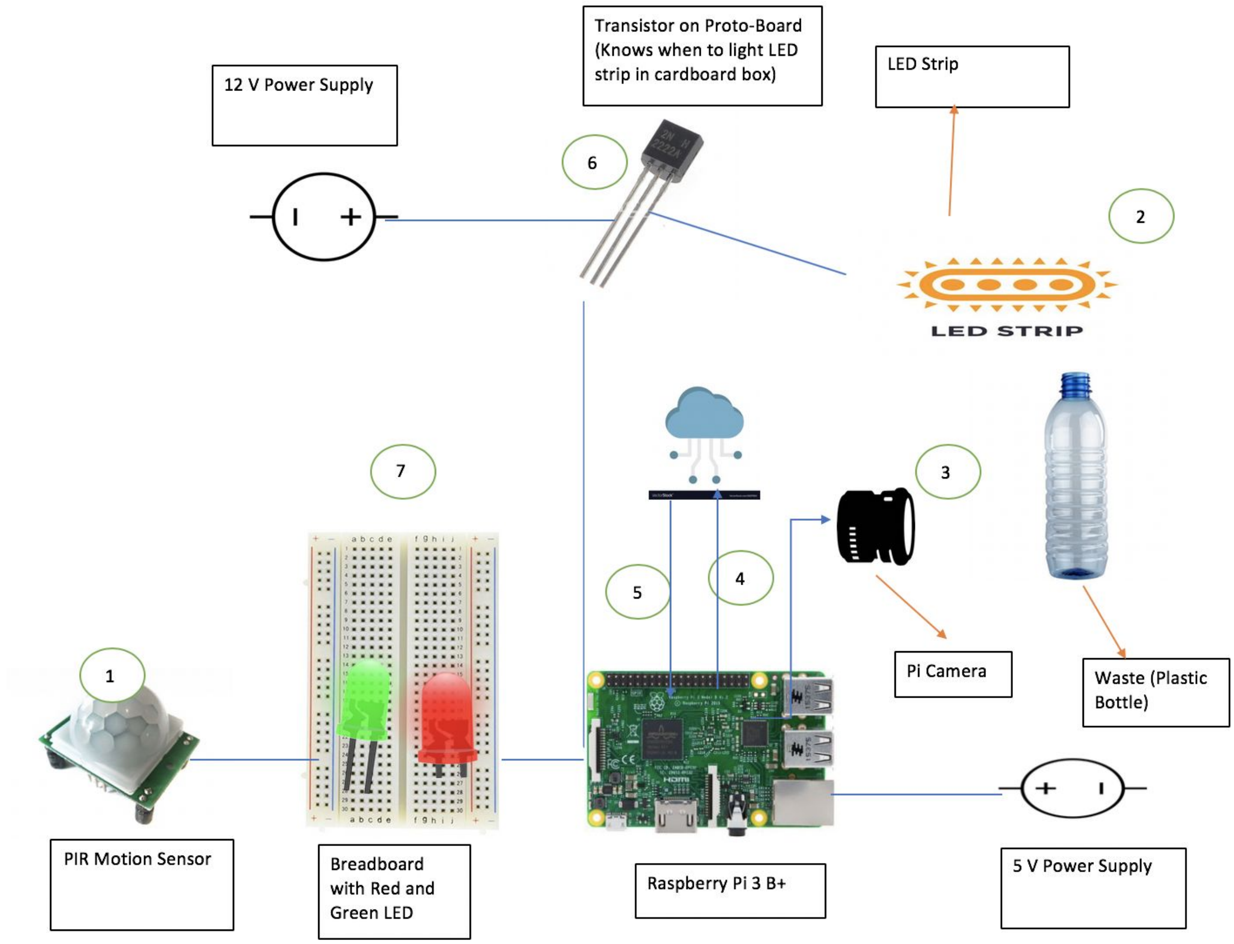

Figure 1: Illustration of Setup and Process Flow

<u>2.1 Hardware Setup:</u>

Ecobin uses a Raspberry Pi 3 B+ microcontroller to relay signals to the LEDs and the camera, receive signals from the motion sensor, and communicate information to the cloud server. The Pi Camera captures the image of item in the bin. To address poor lighting conditions inside Ecobin, the team equipped the prototype with a LED strip with high luminance. Since the RasPi outputs 3.3V, it cannot support the 12V LED strip. Therefore, the LED strip has its own 12V power supply. A BJT transistor was used as a switch for the LED strip, so the RasPi can determine when the strip should be on or off.

The prototype design also has 3 other LEDs: white, red, and green. The white LED responds when a motion has been detected. The red LED will light up if the algorithm detects the object as trash, while the green LED will light up if it detects it as recyclable. The last hardware component is the PIR motion sensor, which has 3 ports: VCC, output, and GND. VCC and GND are used for power supply, while the output port transmits a signal to the RasPi when a motion is detected. The following table shows the pinout instruction between the RasPi, motion sensor, and the LEDs. The camera is already attached to the RasPi.

The diagram above (Figure 1) is different from the one in the testing plan, which originally did not have a connected LED strip to illuminate the given object. The team implemented additional functionality to the setup by soldering and powering up the LED strip shortly before the demo.

Raspberry Pi pin numbers	Usage/Description
2	5V Power -> PIR Vcc
6	Ground -> PIR GND
11	GPIO 17 -> PIR Out
3	GPIO 2 -> White LED
5	GPIO3 -> Red LED
7	GPIO 4 -> Green LED
13	GPIO 27 -> LED strips

Table 1: Hardware Pinout

2.2 Software Setup:

For this prototype, a local computer simulated a cloud instance, which the final prototype would use to communicate with the Raspberry Pi (an Amazon Web Services GPU EC2 Instance with 32GB RAM, 8 CPUs and will cost $3/hour). The Pi constantly waits for object validation data from the server, as well as motion detection information from the PIR sensor. As soon as the trash image is taken, the Pi then sends this image to the local server to let it process through a deep learning model via Keras.

Within this simulated server, a Python script, *keras_test_individual.py*, powered by the Keras API and the pre-trained VGG model, identifies object in the image. (Keras is an open source high level neural network library which allows for efficient and fast prototyping and experimentation.) The server then communicates with the Pi and then a signal is sent to the LED which in turn lights up green(if recyclable) or red(if trash).

2.3 Pre-Testing Procedure:

Hardware Side:

1. Make sure the ports are connected according to Table 1.
2. Check the electric circuit (resistors, LEDs, ground, power supplies, transistor)
3. Prepare a white background for the object since the pre-trained object detection algorithm mostly uses white background. To do this, the team used white paper and cardboard.
4. Position the camera, around 15-20 cm from the object and cardboard.
5. Position the motion sensor, so it won't detect unnecessary motion during the testing phase.

Software Side:

1. [Computer] Enable ssh connectivity with the Raspberry Pi with the command *ssh pi@*ip_address**
2. [Computer]Enable sftp connectivity with the Raspberry Pi with the command *sftp pi@*ip_address**
    i. Make sure that the sftp environment is in the correct working directory. (ie. *cd Desktop*)
3. [Computer] Run the python script, *keras_test_individual.py*
4. [Raspberry Pi] Run the python script, *ecobin.py*

2.4 Testing Procedure:

1. Place an object in front of the cardboard

2. Wave hand over PIR motion sensor, which will turn on the white LED. In addition, this will send a signal to the transistor which lights up the LED strip, and trigger the camera.
3. Image of trash is captured by PiCamera and saved to current directory in the Pi.
4. The image is processed in the simulated server, in this case, a computer.
5. [Computer] - Through sftp, enter the following commands
    i. *get trash.jpeg* ← Sends object image to the computer from the pi
    ii. *put code.txt* ← Sends detection results to the pi from the computer
6. The computer sends the results back to the Pi, the Pi shows the result through the red (trash) and green (recyclable) LEDs.

3. Metrics of Prototype Testing:

3.1 Method of Measurement:

The overall performance of the prototype is measured by qualitative and quantitative metrics. The qualitative analysis, based on a set a predefined criteria, examines how well the hardware and software functionalities are integrated. The quantitative analysis measures the degree of accuracy of the object detection algorithm, an imperative feature of the Ecobin, through a score sheet.

3.2 Prototype Criteria:

The criteria for a successful prototype are as follows:

1. If there is a motion above the PIR motion sensor, the white LED strip will lit up
2. The Raspberry Pi should successfully capture an image and output its name to the terminal.
3. The team wrote a list of common recyclable disposals in a .txt file. The object detection script should identify the object in the container using the VGG model, and search through the mentioned list.
4. If there is a match between the identified object and an item on the recyclable list, the green LED will light up. Otherwise, the red LED will.

5. The prototype design will utilize two power supplies, one for the RasPi (5V), and one for the LED strip (12V). A 12V supply unit powered the LED strip, as it is rated at a different voltage than that of the RasPi.
6. The Raspberry Pi should successfully classify whether an object is recyclable with **75%** accuracy.

3.3 Score Sheet:

The score sheet measures the quantitative success of the prototype. The following table was used during the prototype and marked by the appropriate instructors and GTAs. The prototype is successful if it has a score of at least 75%.

4. Result of Prototype Testing:

The result of the prototype testing is evaluated based on the score sheet. The prototype got scored 90%, based on 10 images. The error was for an apple which was classified as recyclable. This is undesirable as it would be better to have recyclable objects go into the trash, rather than trash go into recyclable. Regardless, the overall performance of the prototype has met the requirements of the score sheet.

Object	**Category**	**Correct? (Y/N)**
Plastic Bottle 1	**Recyclable**	Y
Plastic Bottle 2	**Recyclable**	Y
Plastic Bottle 3	**Recyclable**	Y
Plastic Bottle 4	**Recyclable**	Y
Apple (red)	Trash	N
Apple (green)	Trash	Y
Deformed Paper 1	Trash	Y
Deformed Paper 2	Trash	Y
Orange 1	Trash	Y

Orange 2	Trash	Y
Result →		**90 %**

Reflecting on the aforementioned prototype criteria, the following observations were made:

1. The system was activated by passing a hand over the PIR motion sensor
2. The white LED strip lit up and the captured image was successfully identified and output onto the terminal screen. For example, when a green apple was detected by the system, it output "granny smith".
3. Following the example of a green apple, the algorithm tried to match "granny smith" with items in the recyclable list.
4. Since it is not recyclable, the red light LED activates.
5. The Pi was able to successfully classify it with higher than 75% accuracy as desired. From 10 trial runs, it passed 9/10 tests which leads up to 90% accuracy. (However, a larger data set has to be tested to confirm the number)

5. Evaluation:

The following measures can be taken to improve the product's quality:

1. Conduct more trial runs of equally distributed recyclable and trash objects to ensure higher classification accuracy.
2. Take multiple pictures of same object from different angles (especially for non symmetrical objects) to see how the classification performs. This would indicate the best position to place camera and discover the benefits of having multiple cameras.
3. A pre-trained deep learning Keras Library was used in the prototype for fast prototyping. However, implementing a neural network using TensorFlow would produce a more robust and customizable capability.
4. Use a BU SCC (Shared Computer Space) cloud instance to train and run neural networks for image classifications.
5. Reduce true negatives (object is recyclable but classified as trash) and false positives (object is trash but classified as recyclable). It would be more desirable to have true

negatives rather than false positives, because having trash in the recyclable compartment would just burden the recycling plants and generate additional costs.

6. <u>Conclusion:</u>

The prototype achieved its initial goal by integrating neural network system (software) with the Raspberry Pi and camera (hardware) elements. The prototype correctly identifies the object in front of the camera and also classifies it into either of the two overarching categories: Trash or Recyclable.

Before the next prototype, there are three objectives that the team would like to work on. First, Ecobin would like to integrate its Raspberry Pi system with a cloud instance on the BU SCC. Second, the software team will initialize efforts to build a neural network using Tensorflow. Lastly, the team will provide the Ecobin system with a larger test set. It would be imperative to increase the number of distinct test cases(i.e. from 5 → 30) as well as the number of images per object(from 2 → 4) to obtain a more grounded sense of functioning capability and accuracy. With the implementation of these three functionalities, the team hopes to deliver an improved second prototype.