BinBot

Requirements Specification

REVISION HISTORY

Revision #	Author	Revision Date	Comments
1.0	Michael Savitski	September 17, 2019	Outline
1.1	Sean DiGirolamo	September 19, 2019	Added various sections
1.2	Sean Reddington	September 20, 2019	Additional features and requirements and system block diagram
1.3	Michael Savitski	September 22, 2019	System Overview
1.4	Jose Silva	September 22, 2019	Additional User Story
2.1	Sean DiGirolamo	September 29, 2019	Revision based on comments
2.2	Sean Reddington	October 7, 2019	Revised and added new diagrams, added diagram explanations, logo, formatting
2.3	Michael Savitski	October 11, 2019	Added machine learning diagram with description
2.4	Sean Reddington	October 11, 2019	Finalized initial revisions, replaced Flask App method, updated diagrams, updated feature and requirements, various proofing and formatting

Table of Contents

System Overview

BinBot is a waste-collection robot intended to patrol specific areas such as university grounds, stadiums, boardwalks, or schools. BinBot will identify waste that is laying on the ground and collect it to be disposed of properly. BinBot will have a camera module and on-board microcontroller unit that communicates with a server via wi-fi. To outsource the heavy data processing from the on-site robot; a Linux server will process the images sent by BinBot using data representations created with a deep learning algorithm to identify pieces of waste. The server will then inform BinBot of information regarding the photos, such as if any waste has been identified, and if so, how far it is and how BinBot should navigate to the waste. The waste will then be collected using a mechanical arm and place it in a waste bin attached to the robot. The image feed from BinBot, with additional visual indicators of identified waste, can be transmitted to a mobile application for observing BinBot's progress and success.

One of the key components for BinBot to function as needed will be deep learning software for training a neural network model, so that BinBot's other software will be able to identify waste objects in the images from its camera module. This will entail the use of the OpenCV library for processing images, and the use of the open-source deep learning library TensorFlow. Ideally, this separate software component will be run on a GPU supported computer system. The neural network data model will be trained using hundreds to thousands of images of waste objects, captured in the expected resolution of BinBot's camera. Also included in the training data with these images will be information such as camera height, object size, and object distance, so that BinBot will be able to make estimations to allow it to traverse to the waste objects and successfully collect them. After feeding images to the neural network, the neural network will be tasked with returning whether or not waste is located in the image, and if so, how far the waste is and at what angle it is from the robot.

The completed data model will be applied to software which will run on a server along with a communication socket service, which BinBot will continuously connect to via wi-fi. The images continuously collected by BinBot's camera module will be sent to the server for processing via the OpenCV library, which will use the trained neural network data model to identify waste objects. Once the waste has been identified, as well as the distance and angle, movement instructions will then be calculated and sent back to BinBot's on-board computer board so that it may travel to the nearby waste objects and pick them up. For the purposes of this simple project, BinBot will simply be instructed to turn towards the waste, and travel to it in a straight line.

BinBot's on-board computer will be an Arduino Mega 2560 board with limited functionality. It will be to collect images from the camera module and then transmit them to the data processing API. Receiving data back from the data processing API about waste objects in BinBot's camera's view, the primary purpose of the board will be to operate BinBot's robotic components. BinBot will have robotic treads for traversing across the floor to approach waste objects, as well as a robotic arm for collecting the objects. The software running on the board will need to use the constantly updated information about distance and size of identified objects to operate these components efficiently and successfully collect waste.

Additionally, a companion mobile application will be developed for users of BinBot to install on Android devices. This companion application will be able to receive images from the data processing server which will allow the user to watch a live feed of BinBot's camera view. The server will modify these images visually to have boxes around identified waste objects, and text about BinBot's current progress in its process of collecting the objects such as the estimated distance and status messages like "traversing" or "collecting".

System Block Diagram

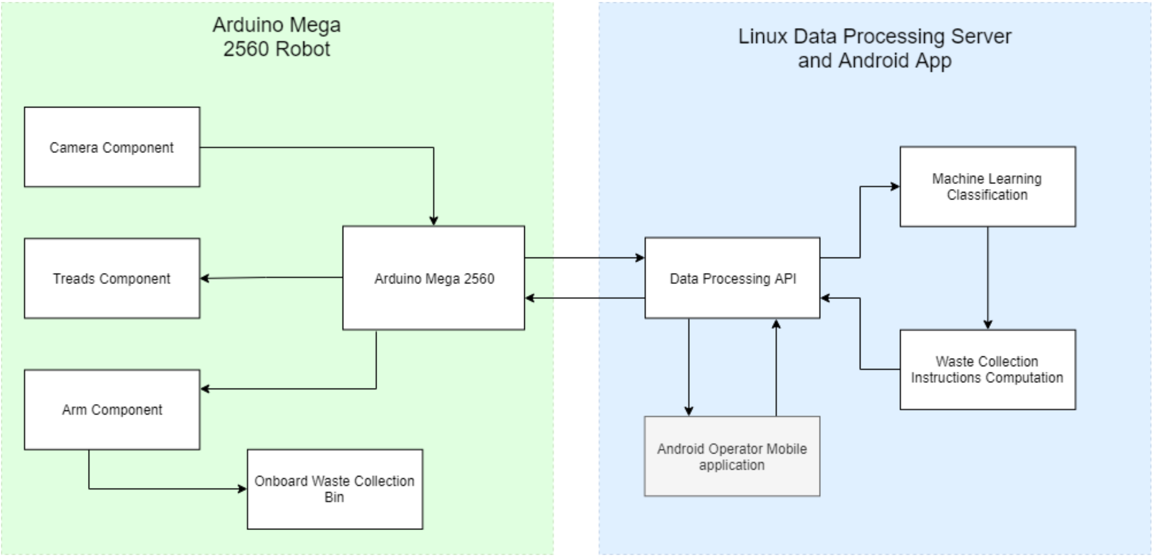
Arduino Mega 2560 Robot
Camera Component
Treads Component
Arduino Mega 2560
Arm Component
Onboard Waste Collection Bin
Linux Data Processing Server and Android App
Machine Learning Classification
Data Processing API
Waste Collection Instructions Computation
Android Operator Mobile application

Fig 1. Block diagram illustrates the flow of data from the Camera feed to BinBot disposing of the waste.

The block diagram shows a high level overview of BinBot project's system architecture. Each of the hardware components on the robot kit will have a corresponding controller interface in the Sketch program loaded onto the Arduino Mega 2560. The Arduino Mega 2560 will exchange data with the Linux server over Wi-Fi via a data transmission interface. The Linux server will be hosting a data processing API that will pass BinBot's camera feed to the machine learning classification processing; passing the results to the Android operator mobile application. If there is collectable waste, the server will then computer instructions for the robot to collect the target waste object and send it back to the Arduino board. The Arduino will then pass the instructions to the tread and arm interfaces to collect the object, placing it into an onboard waste collection bin.

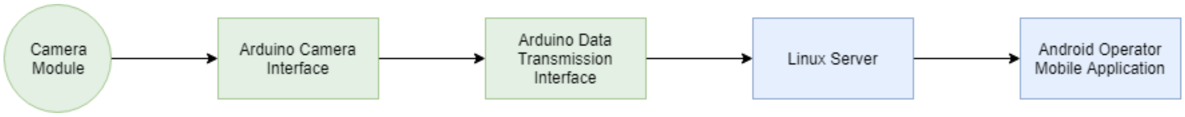
Camera Module
Arduino Camera Interface
Arduino Data Transmission Interface
Linux Server
Android Operator Mobile Application

Fig 2. Diagram displaying interaction between BinBot's Camera Module, Linux Server, and Android Application.

The camera module will capture images at 1 fps via the Camera Interface. This image feed will be passed to the Arduino's Data Transmission interface which will send the images to the Linux server to be processed. Along with the data processing, the image feed will be passed to the Android operator mobile application for monitoring and manual intervention of BinBot during testing.

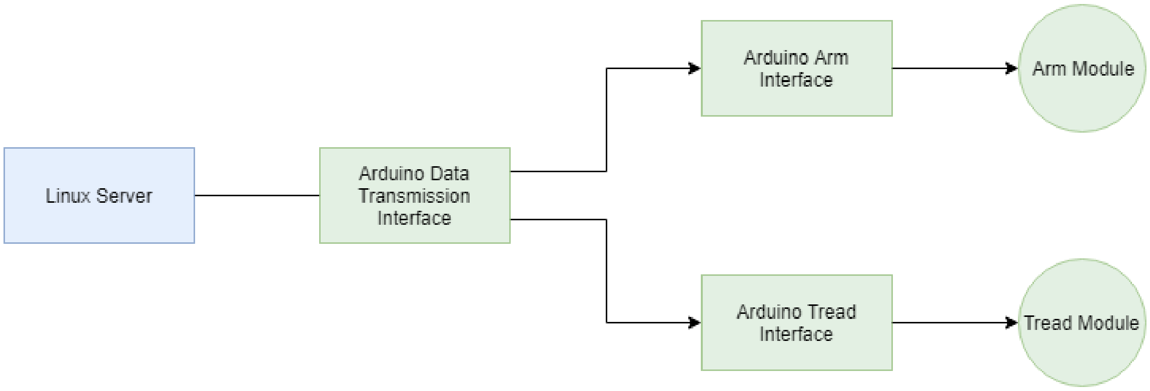

Fig 3. Diagram displaying interaction between the Linux Server and the Arduino's interfaces in order to operate the robot.

The Linux Server will exchange data with the Arduino over Wi-Fi through the Arduino's Data Transmission interface. This includes computed instructions for BinBot to travel to or collect the target object. The Arduino will then pass the instructions to the corresponding interface. These interfaces will provide the functions to move the arm or tread modules.

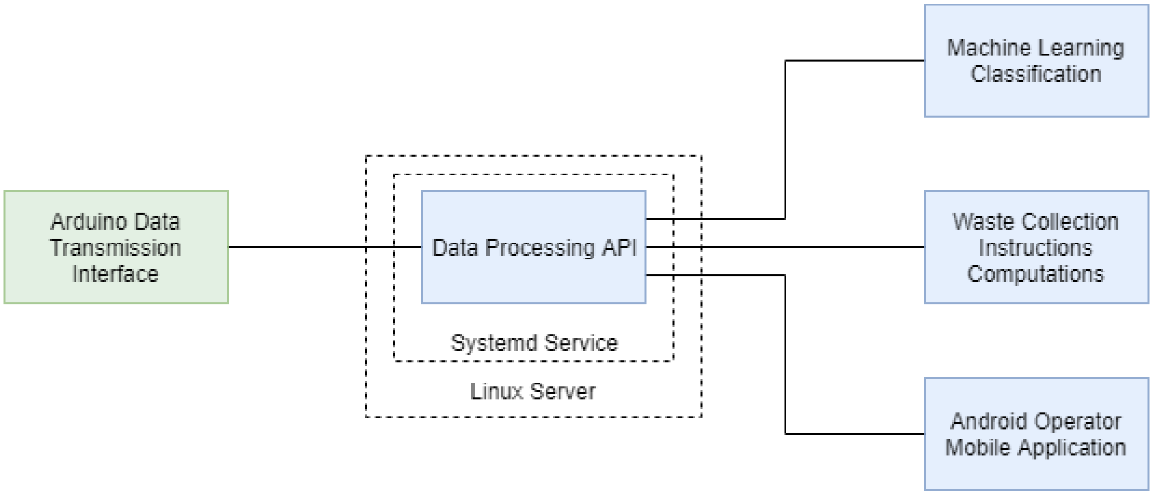
Machine Learning Classification
Arduino Data Transmission Interface
Data Processing API
Waste Collection Instructions Computations
Systemd Service
Linux Server
Android Operator Mobile Application

Fig 4. Diagram displaying the interaction between the Arduino's Data Transmission Interface, Linux data processing server, and Android Application.

The Arduino Data Transmission interface will send the image feed to the Linux processing server. The proposed method is to host a data processing API application that will execute via Systemd services. The data processing API will relay the image feed to the machine learning classification processing. If the machine learning model identifies collectable waste, instructions for the robot kit to collect the waste will be computed. These instructions include navigation instructions for the tread module and collection instructions for the arm module. The image including the results of the object classification will also be sent to the Android mobile application.

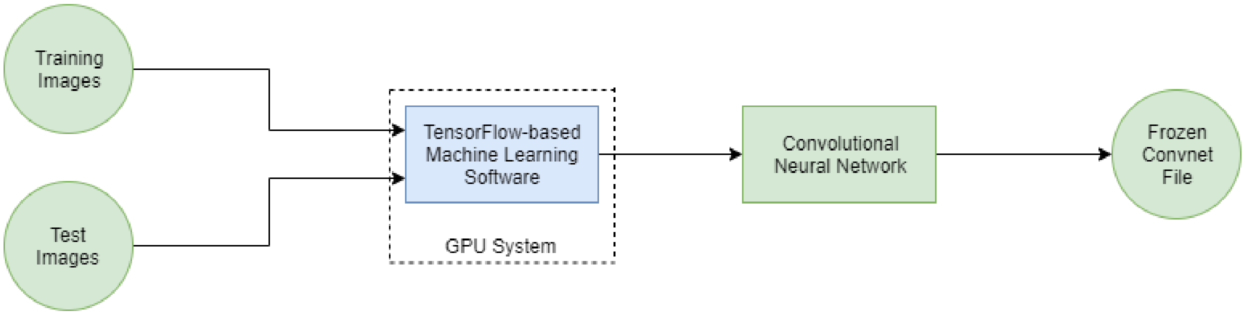
Training Images
Test Images
TensorFlow-based Machine Learning Software
GPU System
Convolutional Neural Network
Frozen Convnet File

Fig 5. Diagram illustrating the process of training the neural network data model that will enable BinBot to recognize objects.

The machine learning algorithm will take as input approximately one thousand images for each trash objects we want to make BinBot's image recognition able to identify. Running on a GPU system and implementing Google's TensorFlow API, the software will train a convolutional neural network data model. There will also be several hundred images reserved as a test data set, to verify the neural network's success on data it was not trained on. This model can then be exported, a process called "freezing", to be implemented in the server-side OpenCV implementation.

Document Overview

The Requirements Specification document defines the features and requirements for BinBot. It includes the following sections:

- General Requirements
- Features and Requirements
- Use Cases

General Requirements

- Nvidia graphics card and machine to run it for machine learning model training
- Android mobile device
- Linux server with wireless communication with robot and mobile app
- Small waste bin
- Xiaorgeek's Arduino Mega 2560 GFS Wi-Fi Video Robot Tank Car Kit with 4 DOF Robot Arm

Features and Requirements

BinBot will have a robot tank car

- Arduino Mega 2560 board with APIs for trend, arm, and camera controls
- Capable of retrieving and disposing of one piece of waste every thirty seconds
- Treads powered by double dc gear motors, planetary gear box, 12W, 1.5W, Degree of climbing: 57°
- Capable of turning and moving at a speed of 1.4 meters per second
- Capable of traversing tile, carpet, and concrete surfaces
- Arm with 4-DOF robot arm and 2-DOF robot pan-tilt platform which will retrieve and place objects into onboard waste bin
  HD camera Two degree of freedom steering Cloud Terrace, Dynamic video resolution can be up to 640*480p, frame up 30F/s which will transmit image feed to data processing server at 1 frame per second with a minimum delay of 2 seconds

Linux server for heavy data processing

- Receive image data from Arduino board at 1 frame per second
- Identify any possible waste from the image feed using machine learning model
- Capable of identifying paper scraps, crushed cans, chip bags, small cardboard boxes
- Determine the location of the target waste to be collected
- Compute instructions for BinBot to travel to and pick up the target waste
- Send mechanical instructions back to the Arduino board
- Send results to mobile operator application for classification confirmation

Android mobile application for robot manual operator mode

- Mobile application capable of receiving images at a minimum delay of 2 seconds
- Allow user to start and stop BinBot robot
- Send user input back to server

Use Cases

Use Case #1

The Dean of Temple wants a way to keep Temple's campus clean as possible.

1. The Dean notices an area on campus with a high volume of waste/liter,
2. The Dean decides to implement BinBot to this area, (either remotely (app remote control feature) or manually (carry it over to the area)),
3. BinBot is activated, *battery as a power source*
4. BinBot uses attached camera to identify objects that are on the floor around it, *camera and sensor features*
5. Once BinBot identifies objects as waste BinBot will traverse to the waste, *trends to maneuver over to the waste*
6. BinBot uses claw attachment to pick up identified waste,
7. BinBot disposes of waste in a bin designed for BinBot, *need to decide on what is going to be used for this with group*
8. BinBot continues its search for waste in the area,
9. BinBot has located and disposed of all the waste in the area,
10. The Dean receives a notification from BinBot that area has been cleared of waste.

Use Case #2

The Mayor of Philadelphia notices large amounts of waste beginning to clutter the city streets. He wishes he had a solution to clean up the waste.

1. The Mayor decides to use BinBot to clean up the city's waste.
2. BinBot is sent to a location in the area, *battery powered*
3. BinBot begins by using its camera to scan the area for waste, *camera scans the area*
4. Once BinBot detects an object, *camera is used to detect an object in BinBot's view*
5. BinBot identifies the object as waste, *identifies if object is waste based on 5 categories*
6. BinBot begins to travel over to the waste object, *trends used to travel over to object and sensor to identify when close to object*
7. Once in proper range BinBot extends claw to pick up waste object and picks up waste, *claw feature used to pick up waste once close enough*
8. BinBot disposes of waste in special waste bin, *designed waste bin for BinBot*
9. BinBot continues its search of waste objects, *continue to search until no more waste in the area*

Resources

BinBot Robot Kit:

http://www.xiaorgeek.com/store/arduino-mega-2560-gfs-wifi-video-robot-tank-car-kit-with-4-dof-robot-arm.html

Similar Projects:

https://www.hackster.io/twob/bluetooth-controlled-pick-and-place-robot-7f7814
https://rootsaid.com/pick-and-place-robot-tutorial/