

Technical Report

Smart Recycler

Ayrton Albuquerque – ayrtonsilva@alunos.utfpr.edu.br
Ian D. A. Queros – ianqueros.1996@alunos.utfpr.edu.br
Miguel Chagas – migueln@alunos.utfpr.edu.br

June, 2022

Abstract

In a society that produces more waste with each passing year, the separation of what is recyclable and what is not can become a problem. When discarding materials, people tend to not pay too much attention if something is recyclable or not and, when not properly segregated, those materials end up in field dumps never to be seen or reused again. Our project comes as a possible solution to this problem at least inside the university environment. The system consists of an automated trash bin for some types of recyclable waste, such as some specific kinds of papers, plastics, and metals. The separation is done using image classification, where a picture of the discarded waste is taken and analyzed by a neural network, and using a set of motors attached to flaps the item is moved to a specific container. As an incentive for recycling, we are also rewarding the users with symbolic credits that could be used in different ways. The present document details the development of the project.

1 Introduction

Waste management has always been a challenge in modern society due to population growth and technological advancements enabling significant increases in waste production. In Brazil, 79.1 million tons of solid waste are generated each year [1] of which 92% is collected, but only 59.5% of those reach a proper destination [1]. When talking about recycling the picture is even grimmer, only 2.1% of all the waste collected is recycled [2] and most of this process is done through manual labor.

In day-to-day life, people segregate waste in trash bins that are usually divided into recyclable and organic types, or in some cases, the recyclables are divided into four categories, and even then not much attention is put into throwing the waste in the correct bin. This makes a second separation process necessary because every material has its own recycling process.

To try to minimize this problem the Smart Recycler was conceptualized, as a system that utilizes an image recognition neural network integrated into an embedded system that controls a sorting mechanism, allowing waste to be thrown at it and automatically segregated to a proper container. The system also provides the user with an incentive for recycling by giving a symbolical credit compensation after scanning a QR code generated and displayed by the system itself with a smartphone.

1.1 Project Overview

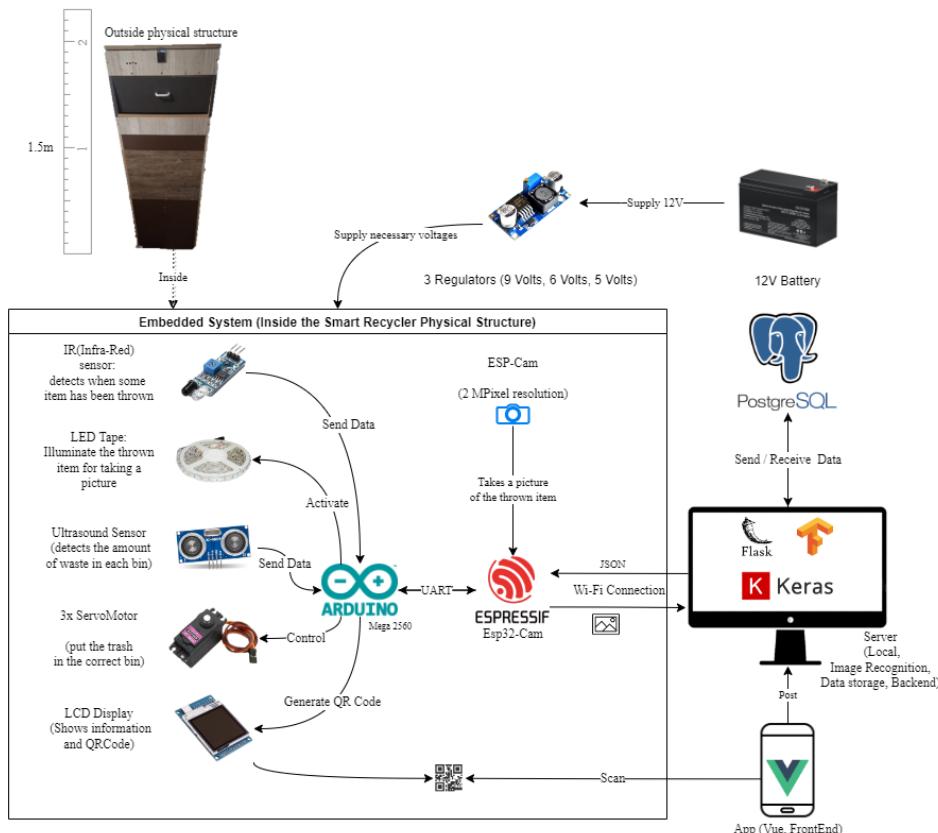


Figure 1: Smart Recycler overview diagram

The project revolves around two main systems, the embedded system consisting of two micro-controllers and the inference system responsible for classifying the waste images. The first micro-controller used is an ESP32-Cam, an Espressif micro-controller with an embedded 2 megapixels camera responsible for capturing the image of the waste item inside the Smart Recycler and bridging the communication between the back end system and the Arduino Mega, the second micro-controller responsible for controlling all the peripherals in the sys-

tem.

The image classification takes place as soon as some item is thrown into the Smart Recycler. To detect this, an infrared sensor was used and after a brief period, an image of the item inside is taken and sent to a back end system that classifies it, returning a JavaScript Object Notation file (JSON) containing the item classification and a redeem code that will later be used to generate a *QR Code*.

When the user scans the *QR Code* he/she will be rewarded with a symbolical credit amount that is stored in a digital wallet. This wallet is part of a Web Application that was developed to help the user maintain its credits and visualize information about the waste recycled presented on section 3.6. The application also allows credit transactions between users.

An item can be classified as one of six classes defined in the project and can be disposed in one of the four containers, those being *soda cans* disposed in the metal container, *juice boxes* and *crumpled paper*, disposed in the paper container, *plastic bottle*, *plastic cup*, and *chip bags*, disposed in the plastic container and finally the fourth container is destined to *others*.

Once an item is classified the recycling process begins by moving a set of three motors divided into two levels in coordination. The first level motor rotates to either left or right and depending on the classification, the second level motor will rotate to front or back. All the information generated during the recycling process is stored in a database and statistics about it can be seen in the web application. An example of each class can be seen in figure 2.



Figure 2: Classes detected

2 Project Specifications

The following describes the requirements, components and tools used in the project.

2.1 Requirements

The project requirements were divided into six areas of development based on the relationship between each technology used. The following tables present the main requirements of the overall project. The complete requirements of the project can be found on the site of the project [3].

Table 1: Mechanical non-functional requirements

Requirement	Description
MS-NFR01	The Mechanical Structure must have one container for each of the four types of waste (plastic, metal, paper and other)
MS-NFR02	The waste containers must have a 25x25 cm base and 60 cm height and the Smart recycler must be at most 150 cm of height

Table 2: Hardware functional requirements

Requirement	Description
HS-FR01	The Hardware system must establish a Wi-Fi connection with the Backend
HS-FR02	The Hardware system must control the flaps of the Mechanical Structure
HS-FR03	The Hardware system must take pictures for material classification

Table 3: Hardware non-functional requirements

Requirement	Description
HS-NFR02	The Hardware system must have a battery as power supply
HS-NFR03	The Hardware system must have three servo motors for HS-FR02
HS-NFR09	The whole process of garbage disposal must take a maximum of 15 seconds

Table 4: Embedded functional requirements

Requirement	Description
ES-FR03	The micro-controller must detect when an object is thrown into the Smart Recycler
ES-FR04	The micro-controller must control the flow of garbage to its correct container

Table 5: Embedded non-functional requirements

Requirement	Description
ES-NFR02	The picture of the thrown object must be taken 5 seconds after the gate's infrared sensors gets obstructed

Table 6: Web Application functional requirements

Requirement	Description
F-FR01	The Web App must allow users to sign in and sign up in the system
F-FR02	The Web App must allow users to visualize their credit balance
F-FR03	The Web App must allow users to send credits to other user

Table 7: Backend functional requirements

Requirement	Description
B-FR02	The system must support users sign in operation
B-FR04	The system must support user balance consults and credit transactions

Table 8: Backend non-functional requirements

Requirement	Description
B-NFR09	The Backend will run in a machine with GPU in the local network of the Recycler, such as the university local server

Table 9: Inference functional requirements

Requirement	Description
I-FR02	The system must detect the object class from the image
I-FR03	The system must provide a level of assurance for its classification
I-FR04	The system must be trained with a custom dataset

Table 10: Inference non-functional requirements

Requirement	Description
I-FR01	The possible inferred classes are: juice box; soda can; crumpled paper; plastic bottle; plastic cup; chips bag
I-FR03	The inference software must take no longer than 2 seconds to classify an image
I-FR06	The neural network will have at most 20% of its inferences be false positives or false negatives
I-FR08	The neural network will be able to recognize only one object per detection

2.2 Components, Materials and Tools

The following subsections describe the components, materials, and tools utilized during the development of the project.

2.2.1 Mechanical Structure

The mechanical structure was built using 15mm thick MDF boards, the only exception being the sorting flaps which were built using 6mm thick MDF to make them lighter and easier to move by the servo motors attached to them. The project was divided into three sections, the container, sorting, and cover. The

cover and sorting sections are unified by 2 hinges between them, leaving only the container section detached from the rest of the structure.

The flaps responsible for sorting the waste are sustained by an 8mm thick iron bar that is attached to a ball bearing in the walls of the structure. This was necessary because the motors themselves were not able to fully sustain the weight of the flap boards. As it can be seen in the section 3.1.

2.2.2 Electronic Components

The embedded system is composed of an Arduino Mega 2560, detailed in section 3.2.1, that controls, receives, and processes the signals from the sensors and motors, and an ESP32, that is responsible for the camera and WiFi connection. The control of flaps of the system is composed of three servomotors mg996r, 3.2.3. The detection of the opening of the gate is done with an infrared LM393 sensor. There are four hc-sr04 ultrasonic sensors , detailed in section 3.2.1, to determine the level of the bins with a red LED each to represent when its bin is full. To illuminate the chamber where the picture is taken a led tape is used, with a relay that controls its activation. For the interface with the user, there is an OLED display 1.6", 3.2.3, that displays the QR code and other messages from the system. All these components can be seen in pictures of subsection 3.2.

2.2.3 Back end

The Back end was written in the Python[4] programming language, chosen due to its ease of reading and writing nature, and implements a REST API using the Flask[5] framework, a library that takes only a few lines of code to implement a production-ready server. It exposes certain routes that allow the web application and the embedded software to communicate with it. The data is received and returned in the JSON format, a widely known format for transmitting data.

To store its data, the Back end uses the PostgreSQL[6] relational database, due to its ACID properties, this database management system was better suited to our needs.

The back end runs in a server running in the same network as the Smart Recycler, the choice of running locally was made due to our scope, university environments, since universities usually have their own servers and powerful GPUs, needed for the inference system, it's cheaper and easier to set up the Back end to run inside the university server.

2.2.4 Web Application

The Web Application was written using VueJs[7], a Javascript framework created to handle the hard parts of writing a single-page application. With it and some of its known libraries like Vue Router and Vuetify, the process of creating a responsive and functional web experience was made possible.

2.2.5 Inference System

The inference system requires three main steps to work: dataset creation, training of the neural network, and the use of the trained neural network model. To create the dataset, VoTT[8] - a visual object tagging tool - was used to annotate our images and separate them into our trained classes. After that, the Python library imgaug was used to implement the process of image augmentation and increase the dataset size. The deep learning API Keras[9] enabled the training of the neural network, keras runs on top of the Tensorflow[10] machine learning library, a widely known library for machine learning projects. To run the network model created the keras API was also used, it allowed loading the neural network model and running the prediction process.

3 Development

This section describes the development and functionalities of the project

3.1 Mechanical Structure

The mechanical structure was first modeled using the *AutoCad* [11] tool and then all the parts necessary to build were produced by a woodworker, starting with the container section. The containers were divided into four parts, each having a 25x25cm base and a total of 60cm height.

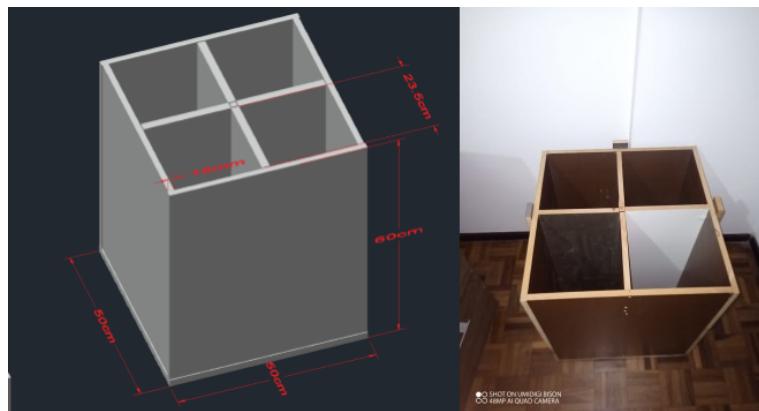


Figure 3: Container section model

The second modeled part was the sorting section. Following the container section constraints, the sorting part has a 50x50cm dimension and 40cm height. It houses two of the three flaps and motors as well as four ultrasonic sensors. To protect the components and wires connecting to the micro-controller a hollow middle section was designed, containing only a hole near one of the walls to allow wires to be passed through.

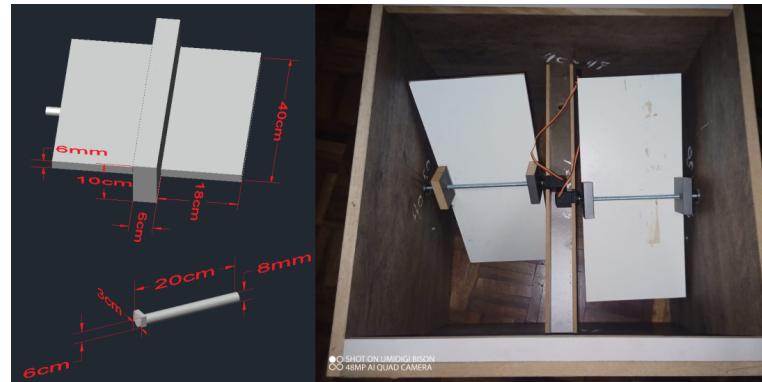


Figure 4: Sorting section model

Lastly, the cover section was modeled. It was designed in a way that separates the electronic components from the waste that would be thrown inside the Smart Recycler.



Figure 5: Cover section model

The final assembled structure can be seen in figure 6:



Figure 6: Assembled mechanical structure

3.2 Hardware Components

The following subsections describe the most important components utilized during the development of the project.

3.2.1 Micro-controllers: Esp32-Cam and Arduino Mega 2560

The ESP32-CAM is a small size, low-power consumption camera module based on ESP32. It comes with an OV2640 camera and provides an onboard TF card slot [12].



Figure 7: Esp32-Cam Module and Arduino Mega

With a wide range of uses, from wireless video monitoring to QR identification, the Esp32 was chosen for this project due to its WiFi image upload capacity. But due to its low number of GPIO ports, it was necessary the utilization of another processing unit.

The Arduino Mega 2560 is a micro-controller board based on the ATmega-2560. The selection of this micro-controller board as one of the main processing units of this project took into account the number of digital input/output pins,

the supported number of PWM outputs, low cost, familiarity, and ease of use of the component by the members of the team[13].

3.2.2 Lead-Acid Battery Zebra Ps4024 and Voltage regulator LM2596

To make the project more portable, we decided to use a battery as a power supply. So it was chosen a rechargeable Lead-Acid Battery with a 12V-7Ah output, capable to supply the whole system for over 48 hours. Due to the voltage output of a little more than 12V, about 12.83V, of the battery, 3 step-down regulators were necessary to safely supply the peripherals. From these three regulators, one is for the servo-motors with an output of 6V, one is for the Arduino Mega with an output of 9V, and a third one is for the other components with an output of 5V.



Figure 8: Lead-Acid Battery

3.2.3 Servo Motor Mg996r and OLED Display 1.6" LCD

This High-Torque MG996R Digital Servo features metal gearing resulting in extra high 10kg stalling torque per cm in a tiny package. This high-torque standard servo can rotate approximately 180 degrees (90 in each direction) [14].

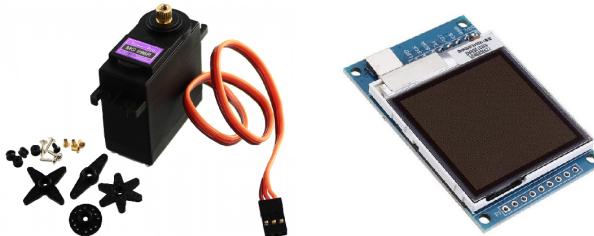


Figure 9: Servo motor mg996r and Oled Display

In this project were used three of such servos to do a precise flap control, with them being controlled by the Arduino mega. To take into account the noise

generated by the servos it was also put a $47 \mu\text{F}$ capacitor between the 6V and the ground.

As one of the interfaces with the user, this display shows to them several messages and the QR code that will be used to redeem the credits in the web app. This display has a 130x130 resolution and an I2C interface. It permits the creation of graphs, in our case QR codes, and texts with ease [15].

3.2.4 Ultrasonic and Infrared Sensors

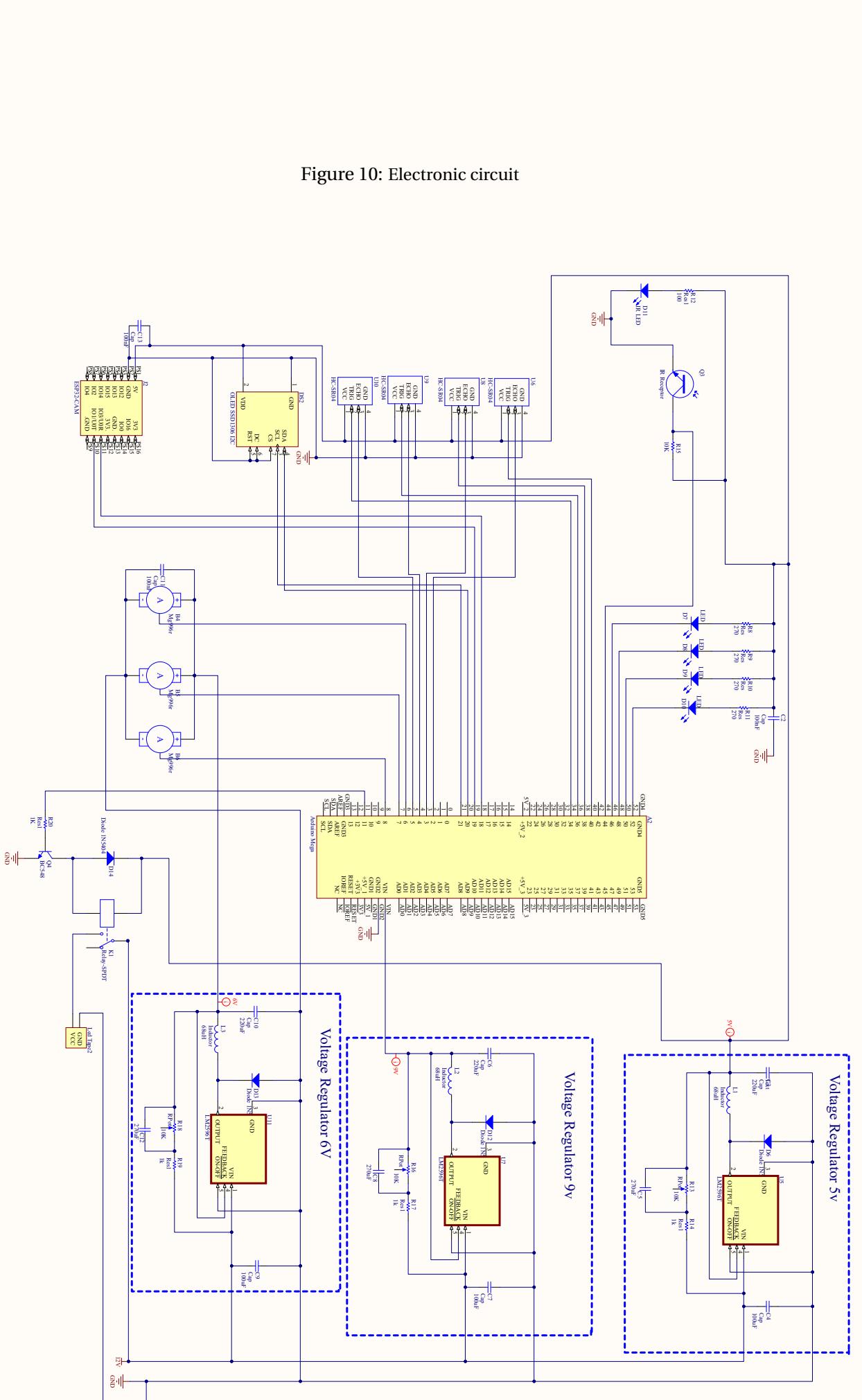
HC-SR04 is an ultrasonic ranging module that provides 2 cm to 400 cm non-contact measurement function[16]. To measure the fill level of each container an ultrasonic sensor Hc-sr04 module is used and to detect when something is thrown inside the Smart Recycler an infrared sensor is used.

3.3 Electronic Circuit

The design began with the individual testing of all the components, making sure of their use and functionality. Then, the complete design of the electronic circuit was developed. After the test of this first design on a breadboard, some interferences were detected. So to resolve this issue three capacitors were added, two between the 5V and ground and one between 6V and ground. Finally, everything was assembled on a standard board.

The ESP32-Cam is powered by the 5V regulator and communicates with the Arduino board via UART. The servo-motors that control the flaps, powered by the 6V regulator, are connected to the PWM pins of the Arduino board. The infrared sensor, used to detect the opening of the gate, is powered by the 5V regulator and directly connected to the Arduino's GPIO. The led tape is directly connected to the 12V battery and a relay module. The relay module is then connected to the 5V regulator and the GPIO pin of the Arduino. The ultrasonic sensors are powered by the 5V regulator with the trigger pin connected to the GPIO and the echo pin to the PWM pins in the Arduino. Each red LED corresponding to one ultrasonic sensor, indicating when the bin is full, is connected to the 5V regulator and the GPIO pins of the Arduino. The OLED display is powered by the 5V regulator and communicates with the Arduino via the I2C pins. The schematic diagram is shown in figure 10.

SMART RECYCLER



Page	1	2	3	4	5	6	7
D							

Table	Page	Number	Revision
	A.2		

Fig. C: Using: simonis, electronic simulation|Drawn By:

Date: 15/06/2022 Sheet of 8

3.4 Firmware

Since the project consists of two micro-controllers two firmware's were created, one for the ESP32 and the other for the Arduino Mega. The ESP32-Cam firmware was the first to be developed and, as pointed out before, its purpose is to take pictures of the waste items inside the Smart Recycler and communicate with the back end. Initially, the micro-controller keeps trying to establish a connection with a specified WiFi network only proceeding when successful.

Once connected, the micro-controller awaits an OK signal from the Arduino Mega through its UART port. When this signal is received a timeout period of 10 seconds is started, activating the embedded camera. When the timeout period has expired the micro-controller will then attempt to take a picture of the item inside it.

With the image captured the ESP32-Cam then tries to establish a connection with the back end so the image can be sent using *HTTP POST* protocols. The received response from the back end is a *JSON* string that is parsed using the *ArduinoJSON* library [17]. Finally, a response is sent to the Arduino Mega in a *JSON* format containing the host IP address, image classification, and redeem code that will be used by the Arduino Mega to generate a *QR Code* to be displayed to the user.

The final ESP32-Cam firmware behavior can be seen in the state diagram in figure 11.

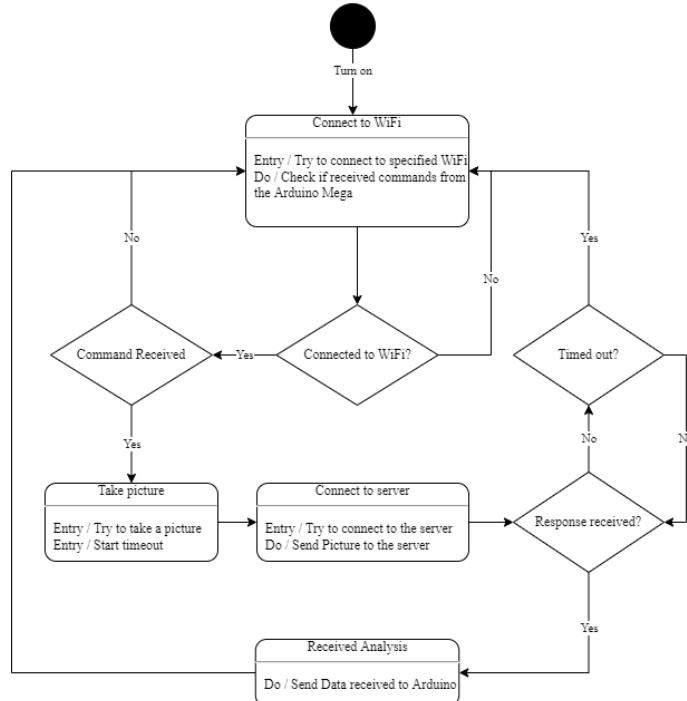


Figure 11: Esp32-Cam state diagram

The Arduino Mega is the second micro-controller and is the one responsible for handling the hardware components in the project. Initially, it keeps reading the infrared sensor until something is detected, starting the recycling process. Once this process begins, the Arduino Mega sends through the UART connection with the ESP32-Cam an OK message, and a 15 seconds timeout is started. If no expected message from the ESP32-Cam is received during this period, the item inside will be recycled to the *Other* container by default.

After receiving a response from the ESP module the Arduino will then generate and display a *QR Code* in the OLED display for the user and based on the classification received the motors will be rotated accordingly. It is worth noting that the motor from the first level must fully complete its movement before any of the motors from the second level begin theirs, so they won't collide with each other.

After properly disposing of the item, the four ultrasonic sensors are measured, updating the fill level of each container. If any of the containers are filled enough so that the amount of waste inside it is within 10cm of the ultrasonic sensor the respective led will be turned on, signalizing that such container is full and no more waste can be disposed of there. If a container is full and an item that should be disposed of in such container is thrown inside the recycler, it will be directed to the *Other* container instead. When the *Other* is also full, the trash bins must be emptied.

The final Arduino Mega firmware behavior can be seen in the state diagram in figure 12.

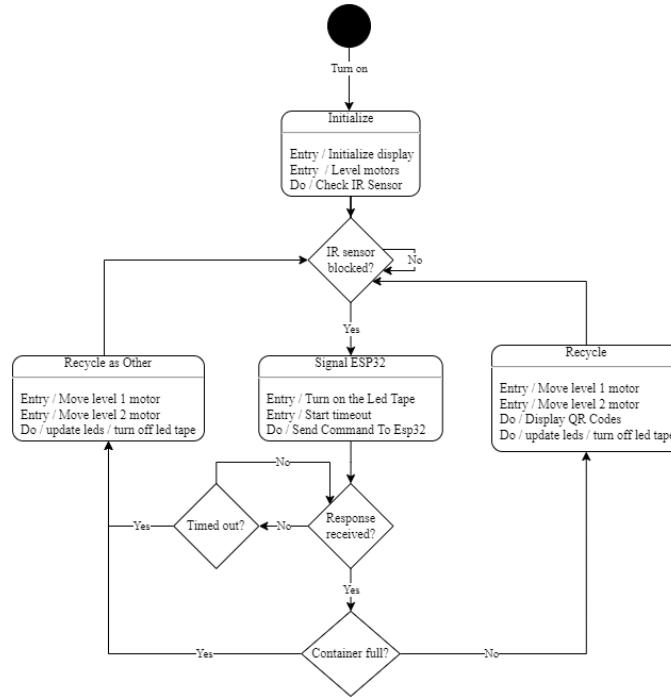


Figure 12: Arduino Mega state diagram

3.5 Back end

With the Flask framework, three different types of routes were created to expose the application functionalities, they were user-related routes, transaction-related routes, and classification-related routes. There are three routes related to users, one GET to acquire user information and two POSTs to sign in and sign up the user. User information is stored in the users database, and its password information is stored encrypted.

There are two routes related to transactions, one POST to send transactions between users and another POST to give the user credits when redeeming a redeem code. Transactions require the username of the sender and the receiver and the amount to send, checks are made to ensure only proper transactions happen.

Finally, there are two routes related to classifications, one POST that receives an image and returns the classification that the neural network model loaded thinks is most likely to be in the image and one GET route to get statistics about all the classifications made. Detailed information about routes and its configuration can be found at the project blog[3].

Finally, to support this routes, the ER Diagram shown in figure 13 was implemented in the database.

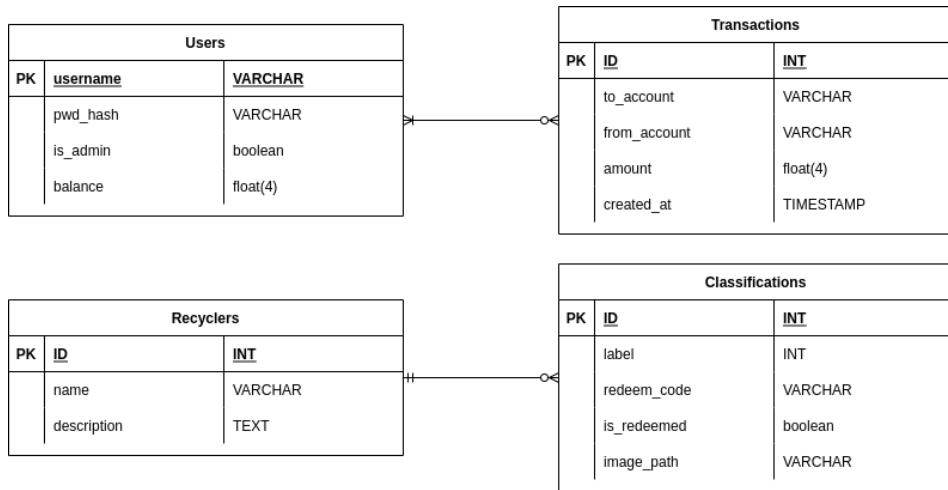


Figure 13: Database Entity Relationship Diagram

3.6 Web Application

The web application was created so the user has an easy and functional way of interacting with the Smart Recycler. A web application was made instead of a native mobile application since it would be much easier for users to redeem credits if they don't need to download a mobile application. The web application is divided into two states: non-authenticated and authenticated. When non-authenticated, the user can sign in and sign up using the corresponding pages. When authenticated, the user can redeem credits, check their account balance, send credits to other users, and check Smart Recycler statistics. The redeem credits functionality works by the user reading the URL contained in the QR code sent by the Smart Recycler and being redirected to the web application, which will store redeem code information using browser cookies if the user is not authenticated and send the redeem credits to the Back end if the user is authenticated. Finally, to send credits the user only needs to specify the receiver and the amount to send. In Figure 14 all web application pages are shown.

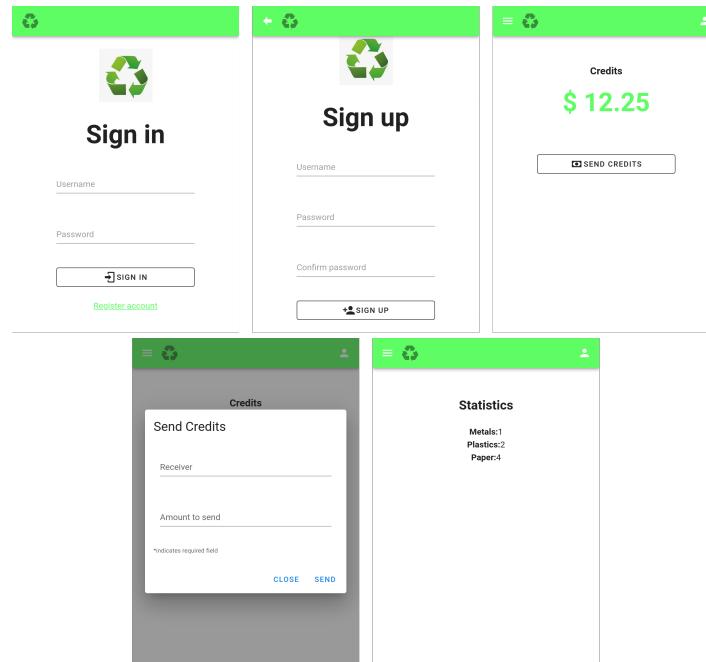


Figure 14: Web Application pages

3.7 Inference System

The inference system was one of the hardest parts to develop in the project since we had a problem specific to it, which was detecting some types of trash thrown in the Smart Recycler, a custom dataset was needed to properly classify our desired objects. To create this custom dataset, we used embedded software that was uploaded in the ESP32 to take images and upload them to a Google Drive folder, allowing us the fast creation of a dataset with images taken directly from inside the Smart Recycler top compartment. Then we used an object tagging tool, VoTT, to annotate the images with the desired classes that we wanted to classify. In Figure 15 an example of the annotation process is shown.

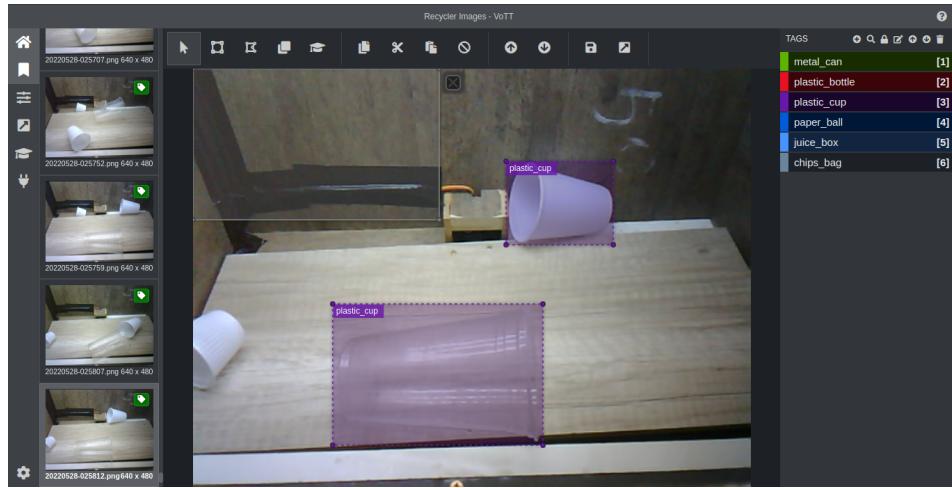


Figure 15: Annotation examples

After annotating and dividing the images into the desired classes, we split the dataset into training, validation, and testing datasets in a ratio of roughly 70%, 10%, and 20% respectively. The training dataset is used by the neural network API to improve its inference accuracy, the testing dataset is used in the training phase to check the model metrics and the validation dataset is used later to test the model with images outside of the training process. To increase the model's ability to detect different situations, we augment the training and testing dataset, where each image generates six other images with certain possible transformations. The augmentations used were the following: Gaussian blur with 50% chance, salt and pepper noise with 25% chance, gamma contrast change with 25% chance, and affine transformations as scaling and rotating the images. In Figure 16, an example of the augmentation is shown.

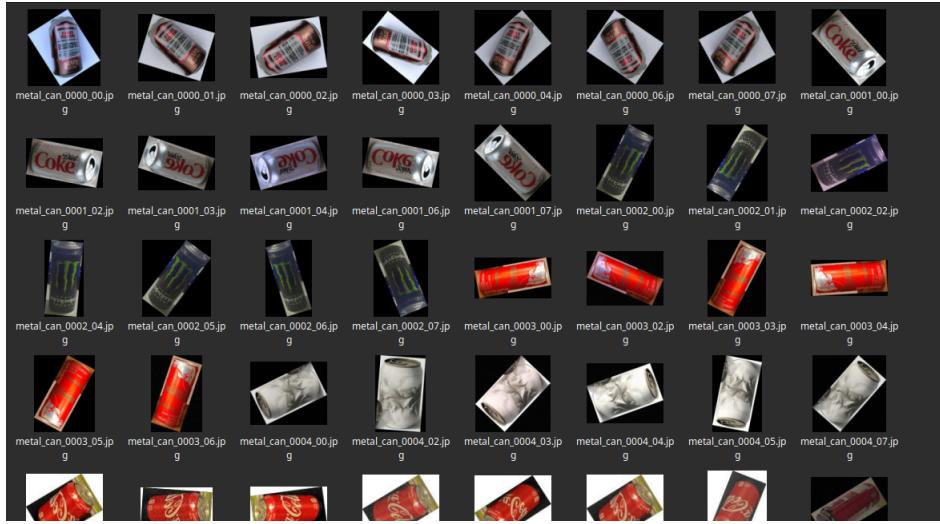


Figure 16: Augmentation output

For the training process, multiple parameters must be chosen, the technique used to speed up network training was the transfer learning, where pre-trained weights are used so only the top layers of the network need to be trained again, using pre-existing feature detectors from last layers, the ImageNet[18] weights were used to transfer learning from. Multiple possible networks can be chosen to create our model, some of the trained networks were: ResNet50V2, MobileNet, InceptionV3, and EfficientNet. The chosen network was the EfficientNetB4 for its moderate size and high accuracy in our dataset. The activation function used was the sigmoid function, since it's an activation function that allows multiple class inferences to be made by each model prediction. After each epoch in the training process, the network is evaluated and accuracy is calculated, if accuracy increases, the model is saved in the file system. Figure 17 showcases the plotting of the evolution in the loss and accuracy metrics in the training and validation datasets.

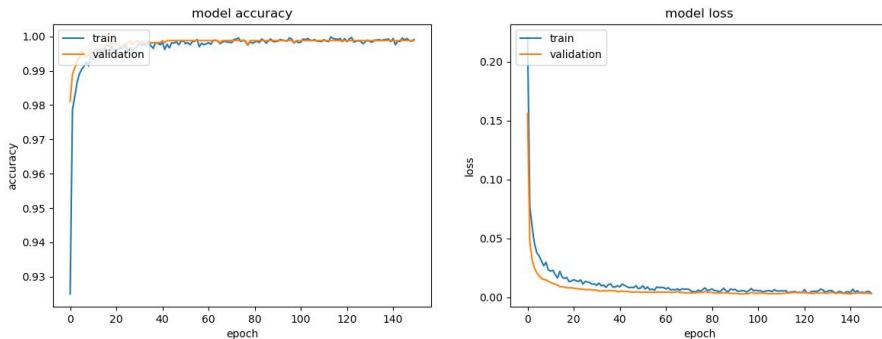


Figure 17: Accuracy and Loss graphs from the trained model

3.8 Budget

Table 11 details the project budget and the amount of money spent on every item. The total estimated cost of the project was R\$ 838,00 but during the development the ESP32-Cam micro-controller was damaged and had to be replaced, increasing the final budget by R\$ 86,00, totaling R\$ 924,00 which was divided between three team members or R\$ 308,00 each.

Table 11: Budget

Item	Qnt	Unit Cost	Total Cost
Arduino Mega 2560 R3	1	R\$ 114,00	R\$ 114,00
Voltage Regulator Xl4015	3	R\$ 14,00	R\$ 42,00
Lead-Acid Battery Zebra Ps4024	1	R\$ 130,00	R\$ 130,00
High Torque Servo Motor Mg996	4	R\$ 50,00	R\$ 200,00
Infrared Sensor Ir Lm393	1	R\$ 12,00	R\$ 12,00
Oled Display 1.6 LCD Pic 130*130	1	R\$ 60,00	R\$ 60,00
Module Esp32-Cam Ov2640	2	R\$ 86,00	R\$ 172,00
Ultrasonic Sensor Hc-sr04	4	R\$ 15,00	R\$ 60,00
Led Tape 3528 White	1	R\$ 32,00	R\$ 32,00
Other materials/components	1	R\$ 150,00	R\$ 150,00
Total	-	-	R\$ 972,00

3.9 Schedule

Table 12 details the estimated and worked hours during the development of the project over the course of nine weeks.

Table 12: Schedule

Deliverable	Estimated	Estimated + 30%	Worked
Blog Page and Project Planning	52	67.6	46
Mechanical Project	33,5	43.55	34,5
Hardware Project	43	55.9	46,8
Software Project	70,5	91.65	74,5
Integration	50	65	48
Overall Integration and Tests	42	54.6	51
Documentation	18	23.4	25,5
Blog Keeping and Meetings	31	40.3	31
Total	340	442	357,3
Delay estimated vs worked	5%	-	-

4 Results and Conclusions

Despite being a mechanically complex project, with several challenges such as movement coordination and image classification, some good results were obtained in the end and the team believes that such a system could be beneficial if implemented in any environment, not only in the planned one. Some major problems presented themselves along with the development, such as the low accuracy of the first trained inference system and the handling of the motors when dealing with heavy objects.

Through the tests performed it was possible to detect some problems that weren't seen in the planning phase. So inevitably some changes had to be made to resolve them. The first time the mechanical structure was totally assembled a problem was noted. The first layer flap and the second layer were too close to each other, and that lead to some waste being stuck between them. To resolve this issue a 10cm separation was added between the top and middle part, which resolved the issue (figure 18).

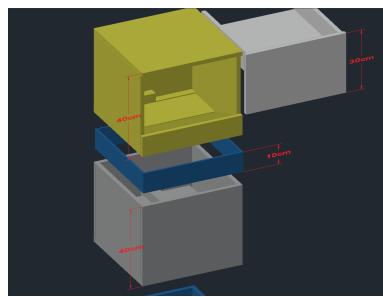


Figure 18: 3D model of the 10cm adjustment

Another problem was the resilience of parts of the servo motors, which were sometimes breaking themselves in their rotating movement. This problem was solved by switching the plastic gears inside the servos with metallic ones, and by applying a smoother sweep, through code, when turning so that they have less pressure in their parts.

Being the source of most of the mechanical and hardware problems, another issue found with the motors was the noise caused in the circuit. This hindrance was neatly solved by adding some filter capacitors in some of the key parts of the system, such as between 5V and Ground, and between 6V and Ground.

The first trained neural network was based on the *ResNet50V2* and despite presenting an accuracy of 86,47% in the training, which was within the required range of 80%, when put to test in a real scenario it performed poorly for a couple of classes. The resultant network was also quite heavy with over 90 MB in size and with a model loss that almost reached 40%.

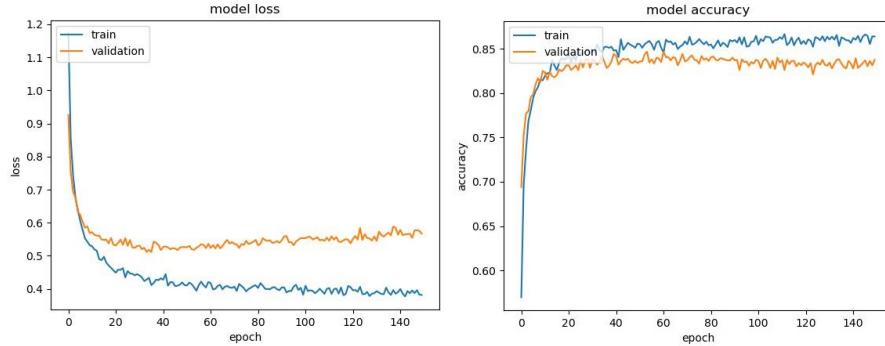


Figure 19: ResNet50V2 loss and accuracy

The second network training was based on the *MobileNetV2* model and the team was expecting good results from this network because of its size. In comparison with the other trained models, this one is very small, with approximately 9 MB in size and despite presenting a similar accuracy and loss as the *ResNetV2* with 86,71% accuracy and 37,28%, when put to the test in a real scenario it performed a little worst.

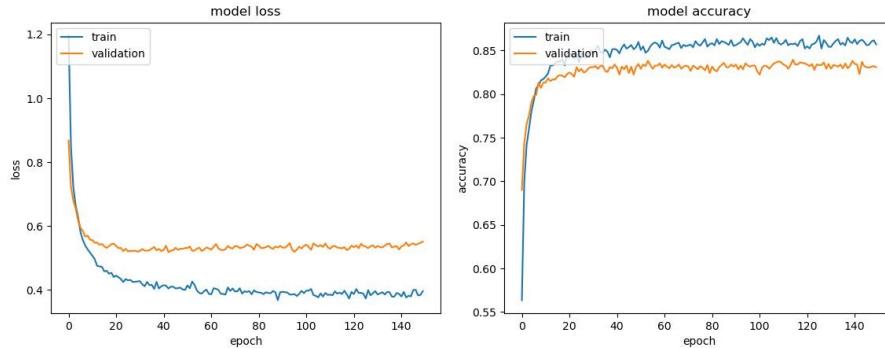


Figure 20: MobileNetV2 loss and accuracy

Thankfully the *EffecientNetB4* model presented a really good accuracy with 99,98% and a very low loss with 0,037% while maintaining a moderate size of approximately 70 MB. The only downside of training this model is the time it takes to finish its training, in our case with 150 epochs taking over 5 hours in a computer with a GTX 1650 TI GPU with a 7.5 compute score. Another good result we had is the image quality from inside the cover as can be seen in figure 2. This was crucial to the success of the project since it affects the training of the network directly.

Despite the issues found along the way, after analysing everything it is possible to say that this project was a success, and our main goal was achieved. A link for the project video can be found in the project site [3].

References

- [1] Yolanda Pires e Nelson Oliveira. Aumento da produção de lixo no brasil requer ação coordenada entre governos e cooperativas de catadores. <https://www12.senado.leg.br/noticias/infomaterias/2021/06/aumento-da-producao-de-lixo-no-brasil-requer-acao-coordenada-entre-governos-e-cooperativas-de-catadores#:~:text=Segundo%20dados%20do%20Panorama%20dos,de%201%20kg%20por%20dia.>
- [2] Gabriella Soares. Reciclagem no brasil atinge apenas 2,1% de tudo que é coletado. <https://www.poder360.com.br/brasil/reciclagem-no-brasil-atinge-apenas-21-de-tudo-que-e-coletado/>.
- [3] Ayrton Albuquerque, Ian Queros, and Miguel Chagas. Smart recycler, a complete view. <https://freezing-gasosaurus-8fa.notion.site/Smart-Recycler-f8e2485b87874f88acfaa625e5c6725d>.
- [4] Guido van Rossum. Python, a programming language. <https://www.python.org/>.
- [5] Armin Ronacher. Flask, web development, one drop at a time. <https://palletsprojects.com/p/flask/>.
- [6] Michael Stonebraker. Postgresql: The world's most advanced open source relational database. <https://www.postgresql.org/>.
- [7] Evan You. Vuejs, the progressive javascript framework. <https://vuejs.org/>.
- [8] Commercial Software Engineering (CSE) group at Microsoft. Vott: An open source annotation and labeling tool for image and video assets. <https://github.com/microsoft/VoTT>.
- [9] François Chollet. Keras: the python deep learning api. <https://keras.io>.
- [10] Google Brain Team. Tensorflow is an end-to-end open source platform for machine learning. <https://www.tensorflow.org/>.
- [11] Commercial Software group at Autodesk. Autocad: Software cad 2d and 3d. <https://www.autodesk.com.br/products/autocad/overview?term=1-YEAR&tab=subscription>.
- [12] Evelta Electronics. Esp32-cam camera module based on esp32. <https://www.evelta.com/esp32-cam-camera-module-based-on-esp32>.
- [13] Arduino Official Store. Arduino mega 2560 rev3. <http://store.arduino.cc/products/arduino-mega-2560-rev3>.

- [14] Eletronicos Caldas. Mg996r high torque - metal gear dual ball bearing servo. https://www.electronicoscaldas.com/datasheet/MG996R_Tower-Pro.pdf.
- [15] BestDualplus. 1.6 "tft oled transflective display módulo 130*130 oled. <https://pt.bestdealplus.com/product/7853222/1-6-tft-ips-spi-oled-transflective-display-modulo-130-130-oled-display-lcd-diy-spi-porta-serial-comunicar-para-arduino/>.
- [16] Powered by HAOYU Electronics. Hc-sr04 ultrasonic sensor distance measuring module. <https://www.hotmcu.com/hcsr04-ultrasonic-sensor-distance-measuring-module-p-58.html>.
- [17] Brian Lough. Arduinojson, the best json library for arduino. <https://arduinojson.org/>.
- [18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.