Bin There Dump That

Authors: Lauren Tan, Jessica Lew, Tate Arevalo: Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—An AI trash can which automatically sorts items into two categories: recyclables or non-recyclables. Users can throw away objects one-by-one into a box, and have them moved into the correct bin by a sliding mechanism. We embedded multiple sensors underneath the platform of the box and mounted a camera on the trash can lid to identify the object's primary material with several sensor classifiers and an image classifier using a neural network model. According to the sensor classifier outputs and the predicted category from the image classifier, an item will be classified as either recyclable or non-recyclable.**

***Index Terms*—Classifier, Jetson Nano, Neural Network, Recycling, Sensor Array, Timing Belt**

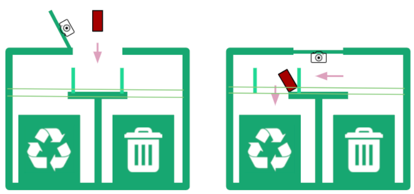

Figure 1: User Flow

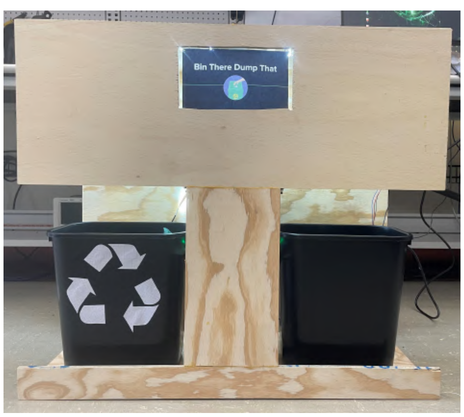

Figure 2: Final Trash Can Exterior

1 INTRODUCTION

The city of Pittsburgh's current recycling rate is a staggeringly low 17%[9]. Compare that to CMU's recycling rate of 35%[3], which is still incredibly low considering that ideally, these rates could be above 50% as seen in some other countries. Bottom line, people do not know how to recycle as well as they think they do. In fact, many people engage in "hopeful recycling", meaning that they recycle what they think should be recyclable, rather than what actually is. Only 31%[5] of people always recycle a recyclable item, meaning that the rate of contamination (non-recyclable items in the recycling pipeline) is extremely high, which reduces the percentage of items in recycling that actually become recycled. Our goal is to help increase the recycling rate on CMU's campus.

To do this, we created an intelligent waste bin capable of sorting waste one item at a time into either the recyclable bin or the non-recyclables bin in under a second and with an accuracy rate of 90%. There are four main types of materials we consider recyclable: metals, glass, PET and HDPE plastics, and paper/cardboard. In terms of similar products on the market, there are few, and some are priced upwards of $1,000. Thus, our trash bin offers a less expensive solution to recycling. Zero thinking is involved in the use of our product as users simply place a waste item into the bin and can walk away. The trash bin automatically classifies that waste item as recyclable or non-recyclable and disposes of the item into the correct bin housed within our product. Gone are the days of debating whether or not certain trash is recyclable, which ultimately leads to low recycling rates and high contamination, as we welcome this easy-to-use and innovative solution.

2 DESIGN REQUIREMENTS

Our main goal is to improve CMU's recycling rate, so our trash can makes recycling easier, more convenient, and more efficient for the average user. Thus, we have developed several accuracy and latency requirements that are critical to the success of our project.

2.1 Classifier Accuracy

First, the accuracy of the overall classifier used to sort items into recycling and non-recycling categories must be at least 90%. This accuracy rate is significantly higher than both the average recycling rate and contamination rate.

The overall classifier combines outputs from the image classifier and sensor classifiers. We tested each individual classifier and then tested the overall classifier to ensure that the resulting percentage of correctly categorized items was at least 90%. To test the image classifier accuracy, only images were used. We used a pre-labeled data set in addition to our own manually collected and labeled data set. To test the sensor classifiers, we placed objects into the sorting box that corresponded to that classifier's target material. For example, metals were used to test the accuracy of the metal classifier. Once we tested the individual classifiers, we placed objects into the sorting box for the overall classifier to categorize as recyclable or non-recyclable so that we could verify the overall accuracy rate.

2.2 Mechanism Accuracy

The second requirement refers to the accuracy of the mechanism that pushes an item into one of the two bins. This accuracy relates to the likelihood of moving an item into a bin after classification occurs. Therefore, this accuracy level needs to be at least 99% in order for the overall system to work as well as possible.

To test this accuracy rate, objects were placed into the sorting box. The rate of successfully placed objects by the mechanism into the specified bin was calculated and compared to 99%.

2.3 Latency

The third requirement for this project is overall latency, which should be less than one second. We define overall latency as the end-to-end processing time of one object. This is measured as the time between an item being dropped into the sorting box of the system and the sliding mechanism returning back to the center of the platform after moving the item into a bin. Thus, this latency includes classifier latency (the time taken to classify an object as recyclable or non-recyclable) and mechanism latency (the time taken to move an object into a bin). This requirement is less than one second, since this is approximately how long it takes for a user to drop a piece of trash into the bin. This also accounts for different scenarios in which users will throw away trash, such as waiting in a line with other people to throw away trash, or simply throwing away multiple pieces of trash by themselves. This timing requirement ensures usability of the overall system with the processing of items one-by-one.

Overall latency was tested by adding the total classification latency to the mechanism latency. Both of these individual metrics were measured separately and averaged over multiple inputs.

2.4 Object Scope

The last requirement ensures that our trash can is able to accept a variety of items including common trash such as water bottles and food containers. Given that our trash can is intended for CMU's campus, we aimed to handle small to medium sized objects of reasonable weight. We define this as objects less than 10x10x10 (WxLxH) inches in size and less than 2kg (around 4.4lbs in weight). This allows us to accept most items that are able to fit into the sorting box, even ones that wouldn't be common around CMU campus, like a brick.

In addition, we are not accepting any liquids or sauces poured directly into the sorting box, as this can interfere with our sensor readings and sensors. Therefore, we are only accepting items that have dry exteriors.

We tested these requirements by placing items of various size, shape, and weight into the sorting box, and then verified that the mechanism pushed that item into the correct bin.

3 ARCHITECTURE OVERVIEW

Please refer to Figure 4 and Figure 17 for our system and software specifications. Figure 4 contains our overall system specification diagram. Figure 17 contains a class diagram which depicts the interfaces that we have defined to facilitate integrating each of our components together.

All of our processing is contained within the Jetson Nano. The limit switch connects to the Jetson Nano via GPIO and indicates whether the trash can lid is closed.

The Jetson Nano receives images from the Raspberry Pi v2 camera, which is connected via MIPI CSI-2 (Camera Serial Interface) and mounted underneath the trash can lid. The inductive and capacitive sensors embedded in the platform underneath the lid send digital output (1/0) to the Jetson Nano. These sensors are connected to GPIO pins on the Jetson Nano. The LED strip is connected directly to an external power supply so it does not need to be connected to the Jetson Nano.

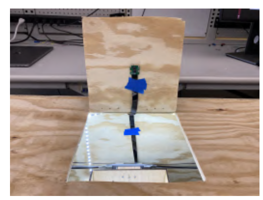

Figure 3: Camera Mounted Underneath Trash Can Lid

The Jetson Nano also controls the motor driver and communicates between the motor controller using two GPIO pins and the `RpiMotorLib` Python library. The motor driver drives the stepper motor for the sliding mechanism.

Once the limit switch is activated, the Jetson Nano begins classification of the next item. The image classifier obtains and resizes images from the camera via the PiCamera library. The image classifier then uses the trained ResNet101 model to classify the object as non-recyclable, or one of the recyclable categories (metals, paper/cardboard, plastics, glass).

Our sensor array feeds data through GPIO pins into the sensor classifier, which we created ourselves. Within the sensor classifier, there are individual material classifiers which output whether or not the inputted item is of that particular material. Each of these individual sensor classifiers is composed of various sensors that are able to detect certain materials. For example, in the metal classifier, inductive sensors were used to determine whether an item was composed of metal or nonmetal because inductive sensors are specifically able to detect the presence of metal.

The image classifier predicted category and the output of the different types of sensor classifiers are used in the overall classifier, which makes the final classification decision of recyclable or non-recyclable. This decision is then sent to the motor controller which prompts the stepper motor of the mechanism to move the sliding belt so that the item is pushed into the corresponding bin.

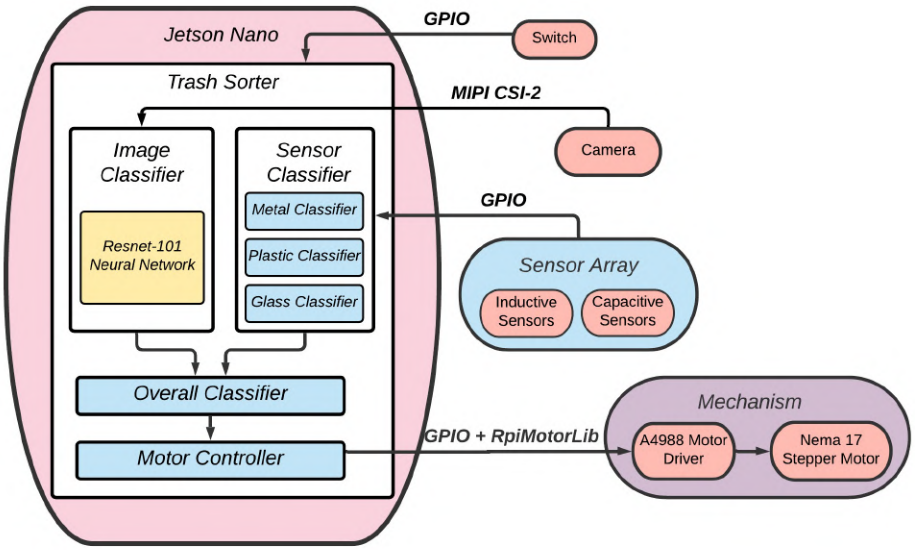

Figure 4: System Specification Diagram

4 DESIGN TRADE STUDIES

Throughout the design process, we discussed several different approaches for our mechanical, software, and hardware components. For each of these decisions, we considered the solution which would be able to meet our design requirements with minimal cost and complexity.

4.1 Mechanism

Timing Belt Mechanism: While developing the mechanism, we considered a variety of different linear motor actuators including a conveyor belt, rack and pinion mechanism, and timing belt mechanism. We ultimately chose the timing belt mechanism because we would not be able to embed sensors onto a moving conveyor belt platform and the cost of the gear rack for the rack and pinion mechanism was much more expensive than the timing belt. During the design presentation, we had intended to use an existing CAD model to laser cut most of the major components of our mechanism including the pulley gear and the motor mount. This method was significantly less expensive than ordering parts, but required time for manufacturing parts and would have to be significantly modified to fit our sliding box design. Since no one on our team had significant mechanical design experience, we instead decided to repurpose a timing belt mechanism from one of our teammates' previous projects which was based on this model from Thingiverse [8]. We still needed to make some modifications such as increasing the size of the timing gear and redesigning the moving platform, but these modifications were much more feasible and only required us to 3D-print the motor and linear rod mounts.

Motor Control: We chose the Nema 17 Stepper Motor and A4988 motor driver because they were among the cheapest models that fit our use case. Based on our design requirements, the mechanism should achieve an approximately 0.5 second latency and be able to move small to medium sized objects. We calculated that the Nema 17 would be able to meet these metrics given its 59 Ncm holding torque and 600 RPM motor speed.

The maximum linear speed of the belt mechanism is dependent on both the motor's speed and the size of the pulley gear which turns the belt. Because we can control the pulley gear size, we only needed to calculate the exact gear diameter necessary to meet our latency requirement.

$$speed = \frac{2 * 10 \text{ in.}}{0.5s} = 40in/s = 1.016m/s \tag{1}$$

$$d = \frac{60 * speed}{\pi * rpm} = \frac{60 * 1.106}{\pi * 600} = 35mm \tag{2}$$

Equation 1 details the linear speed of the belt needed to move the sliding box off of the platform and back in 0.5

seconds. Equation 2 then details the diameter of the gear needed to meet this linear speed given the Nema 17's 600 RPM. From our calculations, we can meet our mechanism latency metric using any pulley gear with a diameter greater than 35mm. As a result, we have selected a 42mm gear.

Then, to verify that the stepper motor would be able to meet our weight requirements, we calculated the minimum torque required to move a mass of 2kg (our maximum object weight).

$$F = m * a = 2kg * 9.8m/s^2 = 19.6N \tag{3}$$

$$torque = F * r = 19.6N * 0.0175m = 34.3Ncm \tag{4}$$

Equation 3 details the force of a 2kg object, and Equation 4 details the torque needed to move that amount of force given the radius of our gear. Since the Nema 17 has a maximum holding torque of 59cm, we should easily be able to support small to medium sized objects.

4.2 Hardware

Jetson Nano: We chose the Jetson Nano over similar platforms like the Raspberry Pi 3 for its powerful GPU and higher performance. The Jetson Nano was designed to run multiple neural networks in parallel and contains a quad-core ARM processor with an 128 core GPU, which allows us to minimize our image classification latency. Initially, we had considered using an Arduino in addition to the Jetson Nano to interface with the motor driver, but decided that step was redundant since the Jetson Nano contains 22 GPIO pins and supports our motor driver. Thus, an Arduino would add unnecessary complexity and latency to our project.

Camera: We chose the Raspberry Pi v2 camera because of its compatibility with the Jetson Nano. The camera connects directly to the Jetson Nano's CSI port and can be easily controlled using the `PiCamera` library. We had considered other cameras with higher quality, but decided that a standard 1080p resolution and 30fps frame rate would be sufficient for our use case. We do not require a higher resolution because we decreased the size of our image for image classification anyway to reduce latency in the neural network. Similarly, we do not require a higher frame rate because we assume that the item is stationary within the bin so we do not need to process a smooth video stream.

4.3 Sensors

We had originally intended to segregate garbage using only computer vision. Although this method was simpler and less expensive than adding a sensor array, we realized that a camera would not be sufficient for classifying visually similar materials such as different types of plastic. Thus, we decided to add a variety of sensors, ensuring that at least one sensor was capable of detecting the following materials: glass, plastic, and metal. While selecting sensors, we consulted research papers and previous projects whenever possible to roughly estimate the sensor's material detection accuracy.

Capacitive Sensor: Capacitive sensors are commonly used for material detection because they are capable of being fine-tuned to detect different capacitance levels. Using capacitive sensors alone we would theoretically be able to distinguish between all of our recyclable materials because each has a specific capacitance range. However, capacitance sensors cost significantly more than any of the other sensors that we were considering such as LDR and IR. This is partly because the sensor output is binary, so we require two capacitive sensors to detect the lower and upper range of capacitance for each target material. In addition to this, we cannot afford capacitive sensors with large sensing ranges, so we would need to purchase many capacitive sensors for full coverage of our platform. Despite all of these drawbacks, we found that alternative sensors such as LDR and IR could not reliably detect our target materials. Consequently, we used capacitive sensors for all of our non-metal sensor classifiers (glass and HDPE plastics), but significantly limited the amount used. We would have also used capacitive sensors to detect paper and PET plastics, but found that those material capacitance ranges were not detectable through testing.

Inductive: To detect metal, we decided that using only inductive sensors should be sufficient based on a research paper which achieved 98% metal detection accuracy using inductive sensors alone [2]. We also considered using capacitive sensors, but due to its previously mentioned drawbacks, opted for inductive sensors as a less expensive alternative. Inductive sensors are still more expensive than some alternatives such as ultrasonic sensors, but we decided that its high accuracy rate warranted the extra cost.

4.4 Sensor Array

We decided to place our sensors in the formation of a sensor array because most of our sensors have small detection ranges. In fact, the sensing range of our inductive sensor is so small (8 mm over the sensor's 0.5 inch contact point) that we would need 400 inductive sensors to fully detect any object on our 10x10 inch platform, which is infeasible given our current budget. After measuring common objects such as water bottles and tin cans, we decided to limit our scope to a 2x2 inch minimum detectable object size because we can rely on our image classifier for smaller objects. Based on this assumption, our inductive sensors can be spaced 2 inches apart, reducing the number of sensors needed from 400 to 16 sensors. Because we require two capacitive sensors at each point to detect the lower and upper range of capacitance, we would still require 32 capacitive sensors for each target material to meet this minimum object size. This means that we would need a total of 64 capacitive sensors to detect both glass and plastic, which would exceed our budget. As a result, we decided that we could reduce the total number of capacitive sensors to 12 and rely on using the image classifier for other materials.

In addition to budget constraints, the number of GPIO pins was a major limitation in the design of our sensor array. The Jetson Nano only has 22 GPIO pins, but given the original design of our sensor array, we would need 28 GPIO pins (12 for capacitive and 16 for inductive). We considered using shift registers but realized that the logic of our system could be simplified to avoid this added complexity. For example, if any one of the inductive sensors detects a metal, the classifier should identify that a metal was detected, even if the other inductive sensors did not detect anything. Because we only need the output of one sensor to be high in order to detect the object, we can wire sensors in parallel which uses only 1 GPIO pin. Overall, we needed 9 GPIO pins (4 for the capacitive sensors, 1 for the inductive sensors, 3 for the motor driver, 1 for the limit switch). We cannot further reduce the number of pins used for capacitance because the capacitance output is binary, so we must be able to detect the lower and upper capacitance range of each target material, glass and plastic. Therefore, we need 1 pin per upper bound, and 1 pin per lower bound.

4.5 Software

Jetson Inference: We had originally intended to train our model on AWS, but decided to train on the Jetson Nano's GPU for simplicity. We used the deep learning library Jetson Inference for both offline training and eventual image classification. This library was designed specifically for use on Jetson platforms and has been optimized for the Jetson Nano. In particular, it uses NVIDIA TensorRT to speed up TensorFlow inference, which helps to minimize our classification latency.

ResNet101 Model: ResNet101 is a built-in model from the `torchvision.models` module in the Pytorch library. It consists of a convolutional neural network made up of 101 layers that has been pre-trained on over a million different images. We decided to use an existing model for our image classifier rather than build a new one from scratch due to the difficulty of determining an object's material from images (i.e. visually similar objects could have different materials), which is necessary in order to correctly classify the object as recyclable or non-recyclable. We had originally considered using ResNet50 because that model has been used to obtain 95% accuracy for garbage classification. However, we found that ResNet50 performed poorly in testing. Switching to ResNet101 did not affect our overall latency, but it did significantly increase our training time. For 60 epochs, the training time increased from around 11 hours to 24 hours, but we decided that this added cost was worth the gain in accuracy. To further increase our image classifier accuracy rate, we also combined images from the Kaggle dataset[7] with our own images. For example, after preliminary testing, we found that plastic bottles were rarely detected by the image classifier, so we specifically added more images of plastic bottles to our dataset.

Training Epochs: Increasing the number of epochs increases our classification accuracy rate but also increases our training time. Figure 5 depicts the validation accuracy for both ResNet50 and ResNet101 as the number of epochs increases. In preliminary testing, we trained our model for only 30 epochs, achieving around 50% validation accuracy. In the final version, we switched to 60 epochs which doubled our training time from 12 to 24 hours but increased our validation accuracy to around 74%. We decided that training for additional epochs would not significantly increase our accuracy because validation accuracy begins to level off around 60 epochs as shown in Figure 5.

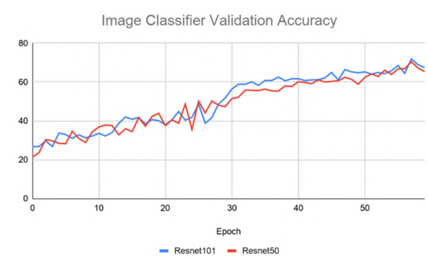

Figure 5: Number of Epochs vs. Validation Accuracy

5 SYSTEM DESCRIPTION

5.1 Image Classifier

The image classifier only uses images as input to classify an item, and has 5 different categories for the output. These categories consist of non-recyclables, and the different types of recyclables: metals, plastics, paper or cardboard, and glass.

An existing dataset[7] from Kaggle, along with our own images, was used to train the model for the image classifier. The model we are using is ResNet101, which is a convolutional neural network with 101 layers. ResNet101 is part of set of pre-trained models from PyTorch's torchvision.models, a machine learning library. The model was trained on the dataset offline using the Jetson Nano's GPU.

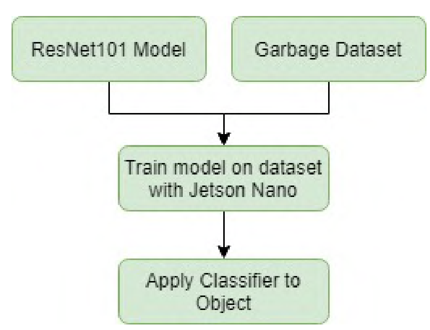

Figure 6: Image Classification Flowchart

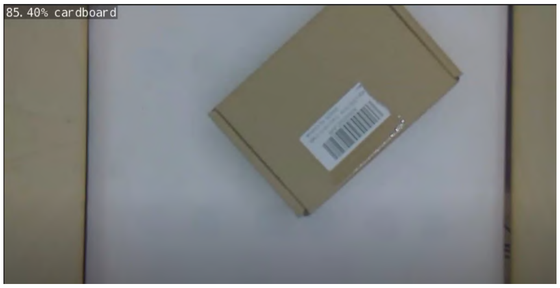

Figure 7: Results of Image Classifier

5.2 Sensor Classifiers

There are three sensor classifiers that correspond to each of the recyclable categories: metals, HDPE plastics, and glass. Each of these sensor classifiers uses a specific kind of sensor to distinguish between each material.

The metals classifier distinguishes between metals and non-metals. This was done using output from a set of 16 connected inductive sensors. The output of each inductive sensors is binary, so each inductive sensor acts similar to a switch and turns on when metal is within the sensing range, and remains off otherwise. By connecting all of these inductive sensors together, the final output of the set of these sensors is "on" if any of the 16 sensors detects a metal, or "off" otherwise. Thus, this sensor classifier for metals has a binary output to distinguish metals and non-metals.

The plastics classifier distinguishes between certain types of plastics (HDPE) that are considered recyclable, and other kinds of plastics that are not. The sensors used to detect these types of plastics are capacitive sensors. The output of each capacitive sensor is also binary, but not all of the capacitive sensors can be connected in the same way as the inductive sensors. For every position on the platform we want to detect plastics, there needs to be two adjacent capacitive sensors. Every pair of capacitive sensors was then connected to the Jetson Nano. Materials like plastics have a range of capacitance values, so we need to know if an object lies within the lower and upper bound of the capacitance ranges for HDPE plastics. Each of the two adjacent capacitive sensors were for these lower and upper bounds, respectively. In addition, since the capacitive sensors can only determine if an object has a capacitance value above a certain threshold, successful plastic detection for objects means that one capacitive sensor had output 1 (for lower bound), and the adjacent capacitive sensor had output 0 (for upper bound).

The glass classifier distinguishes between glass and non-glass objects. Similar to the plastic classifier, we used output from capacitive sensors. Since the output of both of these kinds of sensors is binary, we similarly use a binary output for the glass classifier.

5.3 Overall Classifier

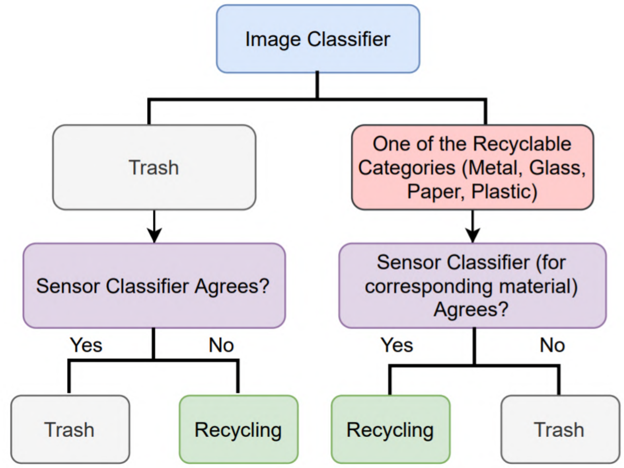

Figure 8: Classifier Flowchart

After combining the output from the image classifier and the outputs of the three sensor classifiers, the overall classifier determines the final category of recyclable or non-recyclable for an item.

The image classifier output is used for most items when the confidence level of the output category is high. This confidence level threshold was determined through later testing of the overall model, and changed for each category.

However, when the image classifier is unsure of the classification of an item (low confidence level), the sensor classifier outputs is used to corroborate or reject the image classifier decision. This depends on whether the image classifier output is part of recyclables or non-recyclables.

If the image classifier output is one of the recyclable categories (metals, paper/cardboard, plastics, or glass), the sensor classifier for that recyclable category is used. The output category from the image classifier and corresponding sensor classifier must agree in order to classify an object as recyclable. This agreement between the image classifier and the appropriate sensor classifier is necessary to reduce the chance of false positives for recyclables, which is more important than false negatives. This is because items that are non-recyclable and sorted into the recycling bin increase the contamination rate, but recyclable items sorted into the non-recyclable bin do not have a similar negative consequence.

Otherwise, if the image classifier output is non-recyclable, the sensor classifier output is used to make the final decision. For example, if the image classifier determines an object to be a kind of metal, but the metals classifier output doesn't detect any metal, the final classification of the item will be non-recyclable. In the other case where the sensor classifier does classify an item as one of the recyclable categories, the final classification of the item will be recyclable.

5.4 Sensor Placement

We created a sensor array, which is a configuration of various sensors in a strategic pattern, because most of our sensors have a very small sensing range and thus would make our sensor classifier extremely unreliable if we just used one sensor.

The grid depicted in Figure 9 shows a bird's eye view of our platform. Our platform is 10x10 inches and 2x2 inches is our minimum detectable object size. So, based on our sensor configuration, we are able to detect an item with some part of the sensor array no matter its placement on the platform. Items smaller than 2x2 inches are primarily handled by our image classifier.

In total, 16 inductive sensors and 12 capacitance sensors were placed throughout our sensor array. Again, 2 capacitance sensors are used in pairs to detect specific materials since one is needed for the lower bound of capacitance for that material and the other for the upper bound. The necessary spacing to avoid interference between sensors has also been taken into account.

Each sensor is mounted underneath the platform through cut-out holes so that the sensors are flush with the platform. Furthermore, as shown in Figure 11, our capacitive and inductive sensors have a built-in contact point as depicted by the orange cap.

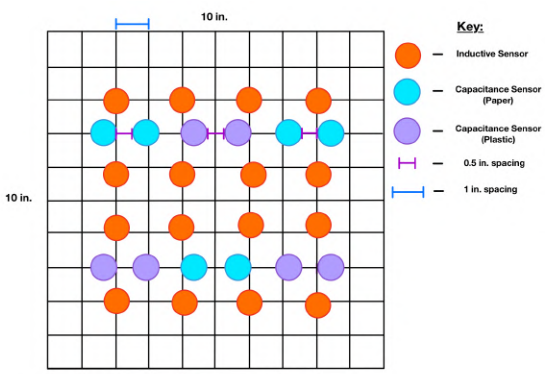

Figure 9: Sensor Array Diagram

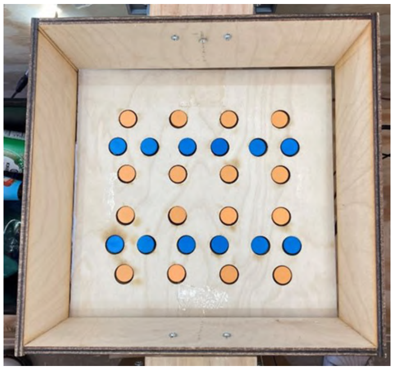

Figure 10: Final Sensor Array

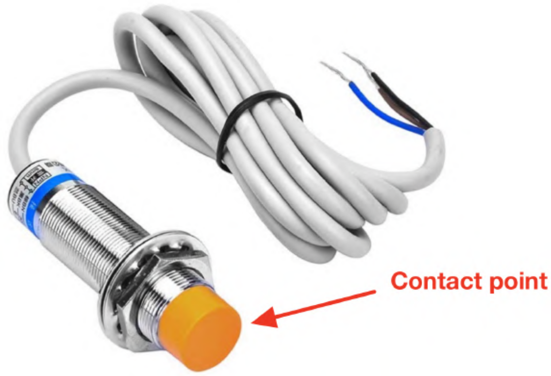

Figure 11: Inductive Sensor

5.5 Mechanism

For our mechanism, we used a sliding mechanism that pulls a box structure across the platform, swiping the item into either the recyclable or non-recyclable bin. As previously discussed, we used a Nema 17 Motor and A4988 motor driver to control our mechanism. Given the Nema 17's 600 RPM motor speed and 59 Ncm holding torque, our mechanism should meet our latency and object scope requirements.

In regard to the design of our sliding mechanism, please refer to Figure 12 and Figure 13. Linear ball bearings attached to metal linear rods make up the fundamental sliding portion of our mechanism. The linear rods are 31 inches in length to allow the box to push trash entirely off of the platform, and they are spaced approximately 11 inches apart to allow room for the box to move in between. The four mounts located on each side of the rods support the linear rods, pulley gears, and stepper motor. These mounts were 3D printed and drilled into the sides of our trash can exterior. To turn the belt, we used two 42mm pulley gears, one fitted around the shaft of the stepper motor and the other held in the opposite mount as shown in Figure 12. The timing belt wrapped around both of these gears and also fastened to the box. The box was constructed from four wooden walls, each the size of our platform (10x10 in). The box was then attached to small shelves protruding from the linear ball bearings as seen in Figure 13. These shelves were directly drilled into the linear bearings.

Once the overall classifier makes its decision, the motor driver drives our stepper motor to turn the gear around its shaft and move the timing belt. Because the belt is fastened to the box, the box is pulled in the same direction as the belt, pushing the item off of the platform and into the correct bin.

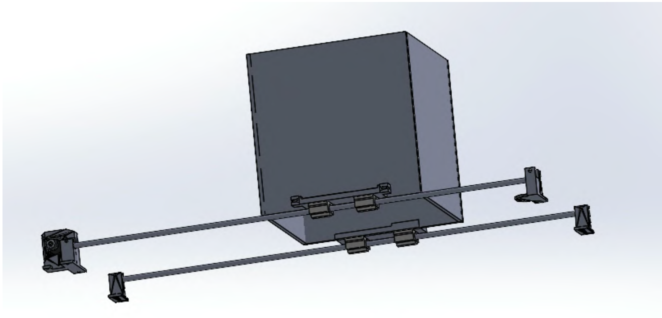

Figure 12: Mechanism CAD Model

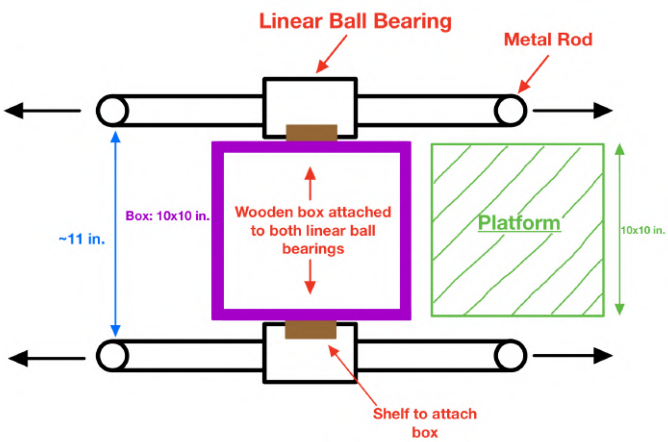

Figure 13: Mechanism Diagram

6 TEST AND VALIDATION

In the testing of Bin There Dump That, our main metrics were latency and accuracy. In the following subsections, we will go over how we tested our product and what our results were.

6.1 Mechanism

Concerning our sliding mechanism, we define the accuracy as the percent of items successfully pushed into the correct bin after classification. We tested a variety of waste items and were able to achieve 100% accuracy for the physical movement of our mechanism. Our original goal was to achieve an accuracy of 99%, so we are happy we exceeded this goal.

The latency of our mechanism was measured by timing how long it took to push an item into the correct bin and return back to the sensor platform. We timed the motion from the instant the mechanism began to move, up until it came to a complete stop. Unfortunately, we were not able to reach our latency goal of less than one second, as our mechanism's average latency was 2.66 seconds. This was the result of a combination of trade-off decisions that we made. Specifically, we decided to go with a greater motor step delay, which increased the accuracy of the sliding box, but decreased the speed of the mechanism. Due to the added weight of the box, which attached to our sliding rails, our motor was unable move at the max speeds it is capable of. In order to improve this latency, a much stronger and more expensive motor would be needed. We did however choose a larger gear size for the mechanism, which helped decrease the latency.

6.2 Classifiers

Material	Sensor	Image	Overall
Plastic	79.6	42.7	89.8
Metal	96	82	98
Glass	76	90	96
Paper/Cardboard	52.2	84.8	91.6
Recyclables	67.1	80.6	93.4
Trash	92	96	89
Overall	67.1	80.6	90.6

Figure 14: Classifier Accuracy Results (%)

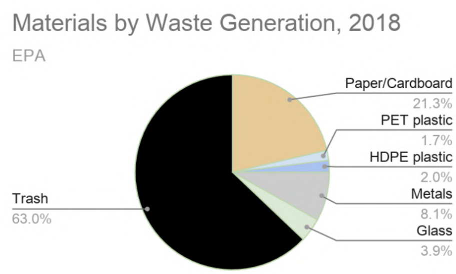

Figure 15: Waste Material Breakdown (%)

Our overall system consists of the following three classifiers: sensor, image, and overall. We tested each of these classifiers separately by placing items into the sensor box and verifying each classifier's predicted classification. As can be seen in Figure 14, our accuracy differs for each material and classifier. For each individual classifier (sensor and image), we conducted 50 trials per material. For the overall classifier, we conducted 100 trials per material. Taking a closer look at some of our significant results, our overall metal classification is extremely accurate at 98%. This is because of how accurate our inductive sensors are in detecting the presence of a metal. Our glass and plastic accuracies are lower because of how far apart our capacitance sensors were placed, due to how few there were for maximizing coverage of the sensor platform. So, depending on how an item lands in the bin, it may not activate any capacitance sensors. However, we did achieve a higher accuracy for glass and plastic after combining sensor outputs with our image classifier outputs.

We also calculated the false positive and false negative performance of our product. We recorded a false positive rate of 11%, which refers to when a non-recyclable waste item was classified as recyclable. In the world of recycling, this is also known as the contamination rate, which negatively affects the proportion of recyclables in the recycling bin that actually get recycled. Since the national contamination rate is 25%, this is a significant improvement. Our false negative rate was 6.6%, which refers to when a recyclable item is classified as trash. Contaminating the recycling pipeline with non-recyclable items is much more costly than putting recyclables in the trash, so the contamination rate is the more important statistic.

In regard to the latency of our classifiers, the classification is extremely quick as the average time it takes for our classifiers to run is 0.117 seconds.

6.3 Overall System

For our overall system, we were able to reach our accuracy goal of 90%, achieving an accuracy of 90.6%. In calculating this overall accuracy rate, we weighted each material's individual accuracy according to the breakdown of trash in 2018 as reported by the EPA, which can be seen in Figure 15. We did this to more accurately represent our system's accuracy rate by reflecting how people throw away waste in the real world.

However, we were not able to reach our overall system latency goal of less than one second. To determine latency, we timed how long it took the classifiers and the mechanism to run. We were able to achieve an average latency of 2.777 seconds. While we missed our target latency metric, we do feel that our current latency is still quite quick and users of the product will not suffer from poor user experience. We also were not able to successfully push our maximum object weight of 2kg (around 4.4 lbs); instead, we found that our product can only support weights up to about 1kg with reasonable latency as shown in Figure 16.

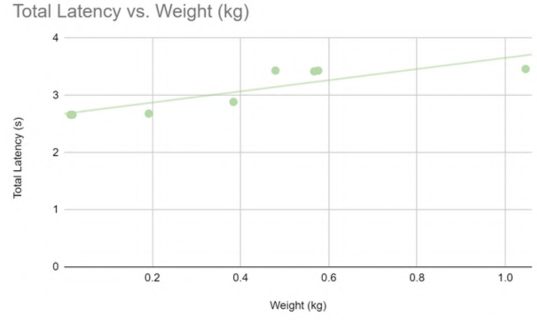

Figure 16: Weight (kg) vs. Mechanism Latency (s)

7 PROJECT MANAGEMENT

7.1 Schedule

In our schedule, we prioritized developing the image and sensor classifiers so that we could better mitigate against unforeseen risks. We ended up allocating more time for calibration due to re-calibration being necessary after attaching paper over the platform, as well as wiring of the sensor array because we needed to switch from a solderless breadboard to a perma-proto breadboard that was solderable. Integration of parts also took longer than expected since our original Jetson Xavier NX stopped working halfway through the semester, so we needed to start from scratch with the Jetson Nano afterwards. The exterior also needed to be re-built due to wrong measurements, so that also added to integration delays. The full Gantt Chart can be found in Appendix B.

7.2 Team Member Responsibilities

Our project is divided into two main categories: mechanism/hardware and classifiers. Lauren's primary responsibility was calibrating the capacitive sensors for detecting glass and HDPE plastics, and wiring our sensor array to a breadboard. She also helped re-train the final version of the ResNet-101 model for our image classifier, and improve the image classifier performance by tuning the confidence levels used after testing. Jessica worked on building the image and sensor classifiers, as well as training our initial and final versions of our ResNet model. Jessica also integrated all components with the Jetson Nano, and designed the mechanism. Tate's primary responsibility was assembling the mechanism and building the trash can exterior. All of us collected objects used for sensor calibration and subsystem testing. Tate did the majority of testing for the image, sensor, and overall classifiers, and Jessica and Lauren worked on the latency testing.

7.3 Budget

The budget can be found in Appendix C. Since we reused parts for our mechanism and didn't end up using some other parts, there is a large discrepancy between the cost used by our budget and the total cost needed to recreate the project from scratch. We have separated the cost from our budget and the cost of recreating the project into the columns "Total Spent" and "Total Cost," respectively. The parts that were purchased but not used in the final version of our product have been grayed out.

7.3.1 AWS

We want to thank Amazon for providing AWS credits for our use in this project. We used $0.13 for trying out AWS instances for training.

7.4 Risk Management

Our main risk factor was classifying edge case objects. In particular, items placed at different angles or items composed of multiple materials are more difficult to classify. This risk can be mitigated against by adding more training data and adding different types of sensors. To account for items placed at different angles, for instance, we included images of items at various angles in our image classifier training set. To account for items composed of multiple materials, we assume that the item has a primary material and we classify based on that primary material. For example, although water bottle lids are not recyclable, we still categorize the entire water bottle as recyclable. Because we anticipated that other edge cases with regards to our classifier would appear throughout the semester, we prioritized classifier testing in our schedule. This gave us enough time to order more sensors such as the ultrasonic sensors and collect more training data as needed.

With respect to our sensors, our largest risks were their accuracies and small sensing ranges. Although we had researched each sensor's approximate accuracy at detecting its target material, it was difficult to estimate their accuracy levels without further testing. To mitigate against this, we prioritized testing our sensors so that we could quickly determine which sensors might need to be replaced. For example, we had expected the IR sensor to not detect glass because infrared rays do not pass through glass. However, while testing, we found that the results were more erratic and were able to order ultrasonic sensors as a potential replacement. We later found that ultrasonic sensors could also not reliably detect glass, so we decided to use capacitive sensors in the final version.

In terms of small sensing ranges, our inductive and capacitive sensors have around an 8mm and a 10mm sensing range, respectively. As a result, we have designed a sensor array such that a normally sized object should have contact with at least one of each type of sensor. If the object is abnormally small, then we rely on our image classifier to make the final classification decision instead. We also left room in our budget so that we could order more sensors for our sensor array if necessary during the semester.

Furthermore, we mounted an LED strip inside the trash can so that our image classifier could still function in low light conditions.

8 ETHICAL ISSUES

Our product uses a camera mounted underneath its lid, which has the potential to be abused by people who use the camera to take pictures of people or other unintended objects. The camera takes a picture once the switch is pressed, so people could position our product to use the camera for their own purposes. This would negatively affect everyone's privacy.

Users have access to the internals of our product when they open the lid, so it is possible for them to put their hand inside while the mechanism is pushing the box off of our sorting platform. This may cause injury if such a person does not remove their hand from our product, since the box would collide with their hand.

In addition, users with physical disabilities that have trouble with fine motor skills could find it difficult to use our product if they cannot open the lid. This would contribute to unequal access in using garbage cans.

To mitigate the above risks that come with having an open lid, we could permanently attach the lid to our product, making it impossible to use the camera to take pictures of people or use it for other unintended purposes. The lid would then become part of the roof of our product, and the camera would stay pointed downwards at the sorting platform. To accommodate this change to the lid, we could cut out a hole on the side of our product for users to put trash into instead of through the top of our product.

There is also the potential for our trained model to not perform as accurately for different parts of the world, since the model is only trained on certain items in our dataset. For example, different objects are more commonly thrown away in different areas or countries. To improve usability for this reason, we could add a much greater variety of items to our dataset.

9 RELATED WORK

There are several projects related to automatic trash sorting. These projects tend to use either computer vision or sensors for classification and are able to achieve reasonable accuracy. Researchers from MIT have used soft robotics to create a trash sorting robot based on touch [4]. The robot can detect size and hardness of the material by squeezing the object. On the other hand, Oscar the AI trash can uses solely computer vision for classification[1]. Similar to our project, Oscar is intended for consumer use and can only sort one item at a time.

While initially developing our project, there were several projects that heavily influenced our decisions. For the sensors, we used the results of "Design and Development of the Trash Splitter with Three Different Sensors" to narrow down which sensors to consider for material detection[2]. We were also inspired by the mechanism in the "Sorter Bin" project which used a stationary platform embedded with sensors and a sliding box to push the trash off of the platform[6]. Unlike the "Sort Bin" project, however, we used more sensors and computer vision to aid our classification. We also used a belt driven mechanism rather than a ball screw mechanism to decrease mechanism latency.

10 SUMMARY

Our system was able to meet all but one of our design specifications. It achieved at least 90% accuracy in classifying objects correctly, ultimately reaching 90.6% accuracy. Our product was also able to meet our 90% accuracy requirement for our mechanism in moving classified objects to their correct bin since it has a 100% accuracy rate.

Our end-to-end latency requirement of less than 1 second was not met, however, since our product takes 2.78 seconds on average to sort each object. Our other design specification of maximum object weight was also not met, since objects weighing less than 2 kg (around 4.4 pounds) were not able to be successfully able to be pushed into one of the bins; instead, we were only able to support objects weighing up to around 1kg.

To improve our system performance if we had more time, we would re-train our image classification model with many more of our own images (hundreds or even thousands). We would also purchase more capacitive sensors to spread out more evenly on our sensor platform, since this would improve the sensor array's ability to detect glass and HDPE plastics. Additional capacitive sensors could also be used to detect materials like cardboard, which would also improve our sensor classifier's performance. Finally, to improve our mechanism, we would purchase a stronger motor that could move 2kg of weight at 600 RPM. The Nema 17 motor that we purchased can handle our weight and latency metrics separately, but slows down significantly for heavier loads.

We learned several lessons while developing this project. First, we should have completely planned out our design before beginning to build. In particular, we began building the trash exterior before finalizing the mechanism so we did not fully account for the height of the mechanism. We also failed to account for the minimum distance between the camera and our sorting platform that was necessary in order to take a bird's eye view picture of objects. Consequently, we had to completely reassemble the exterior so we were not able to integrate the camera with the exterior until much later than expected. Secondly, we should have left more time for testing in our schedule so that we would have more time to detect and handle edge cases, as well as handling unexpected delays in the creation of our product. At the beginning of the semester, we tended to underestimate how much time menial tasks like taking pictures, labeling images, and testing the system could consume, and didn't account for parts getting delayed or breaking, which happened during the semester.

References

[1] Aiman Adil. *AI-Generated Oscar: Trash-Sorting Robot to Achieve Zero Waste.* URL: `https://medium.com/illumination/ai-generated-oscar-trash-sorting-robot-to-achieve-zero-waste-72531ebacb58`.

[2] Despaleri Angin. "Design and Development of the Trash Spliter with Three Different Sensors". In: *IOP Journal of Physics: Conference Series* (2018).

[3] CMU. *Recycling and Waste Management.* URL: `https://www.cmu.edu/environment/recycling-waste/index.html`.

[4] Adam Conner-Simons. *Robots that can sort recycling.* URL: `https://news.mit.edu/2019/mit-robots-can-sort-recycling-0416`.

[5] Covanta. *Covanta Survey: Americans Don't Know How to Recycle.* URL: `https://www.waste360.com/recycling/covanta-survey-americans-don-t-know-how-recycle`.

[6] Maxime Decleire. *Sorter Bin - Detect and Sort Your Trash.* URL: `https://www.instructables.com/Sorter-Bin-Detect-and-Sort-Your-Trash/`.

[7] Kaggle. "Garbage Classification Dataset". In: (2018). URL: `https://www.kaggle.com/asdasdasasdas/garbage-classification`.

[8] DIY Machines. *Robotic Bartender.* URL: `https://www.thingiverse.com/thing:2478890`.

[9] Ashley Murray. *Single-stream recycling: A system that does not work.* URL: `https://newsinteractive.post-gazette.com/recycling/`.

Appendix A

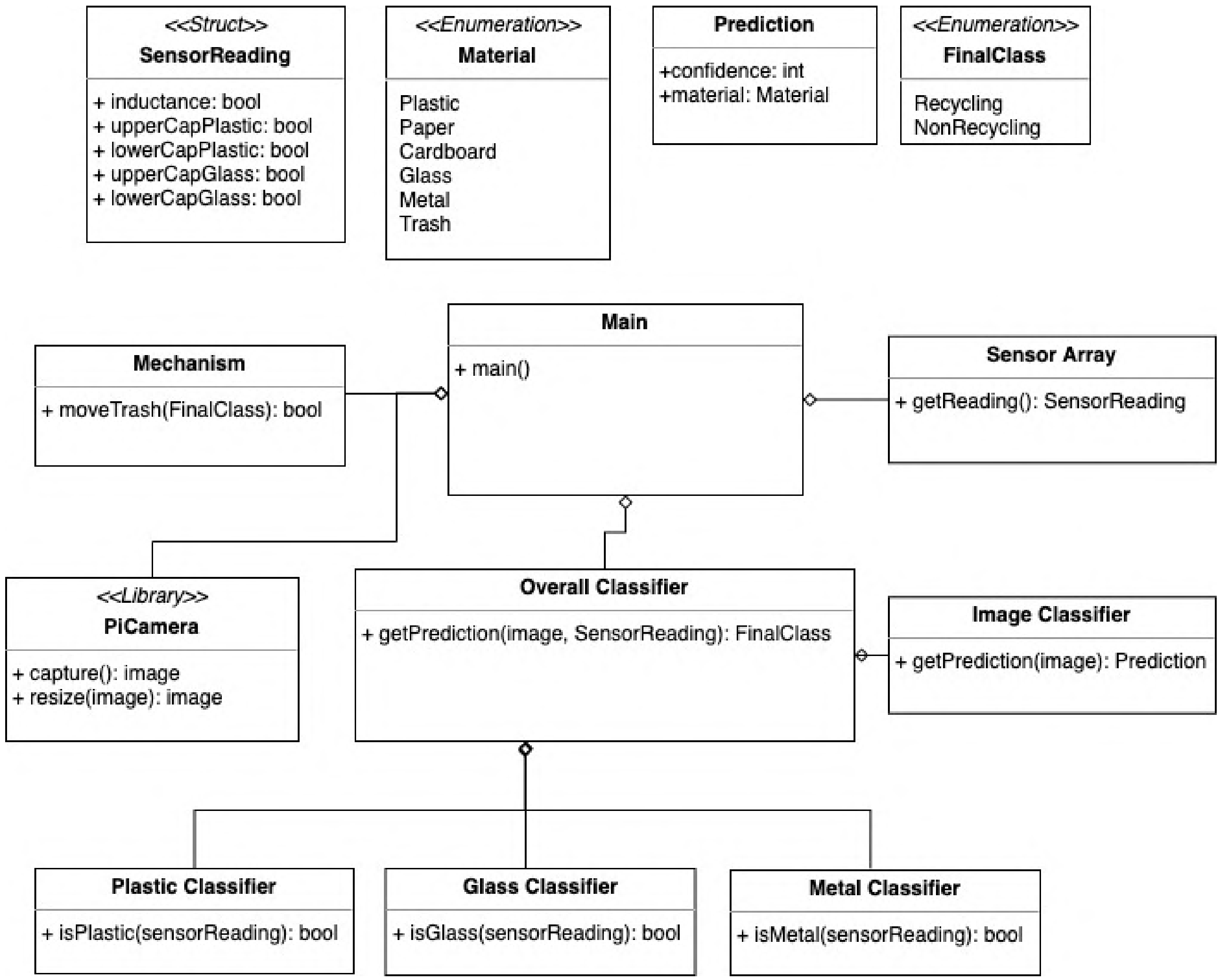

Figure 17: Class Diagram

Appendix B

	TASKS		March					April				May	
		Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Finals
		2/22	3/1	3/8	3/15	3/22	3/29	4/5	4/12	4/19	4/26	5/3	5/10
	Milestones	Proposal Presentation		Design Presentation	Design Report				Interim Demo			Final Presentation	Final Report/Demo
1	**Mechanism**												
	Finalize Mechanism Design	JT	JT										
	Order Mechanism Parts		T	T									
	Build Mechanism				JT	JT	T	T					
	Connect motors to Jetson Nano						J						
	Build sorting box					T	T						
	Build infrastructure					T	T						
	Integrate mechanism + infrastructure							T	T				
	Test Mechanism							JT	JT				
2	**Sensor Classifier**												
	Order Parts	T			L								
	Collect sensor data & test sensors			All	All								
	Build sensor array				All	L							
	Integrate sensor array with Jetson						J						
	Calibrate sensors						L	L					
	Build model for sensor classifiers						J	J					
	Test sensor classifiers							L	L	T			
	Improve sensor classifiers										L		
3	**Image Classifier**												
	Find datasets		J										
	Research Models		LJ	LJ									
	Build Model				J								
	Train model on existing dataset					J							
	Collect Images of real objects								T				
	Train model on dataset + added images							J	J	LJ	LJ		
	Manually test classifer (images of real objects)								J	LJ	All	All	
4	**Integration**												
	Integrate camera with box						T						
	Integrate sensor with image classifier						J	J					
	Test overall classifier (images + sensors)								LJ		LT		
	Test overall (classifier + mechanism)									All			
	Improve overall classifier										LJ	All	
5	**Milestones**												
	Design Presentation		All										
	Design Report			All	All								
	Final presentation											All	
	Final report / demo												All

Figure 18: Gantt Chart

Appendix C

Item	Quantity	Cost	Total Cost	Total Spent	Source
Jetson Nano	1	0	0	0	Personal
Jetson Xavier NX	1	0	0	0	Inventory
IR Sensor	1	8.68	0	8.68	Amazon
LDR Sensor	1	5.88	0	5.88	Amazon
Ultrasonic Sensor	1	7.31	0	7.31	Amazon
Load Sensor	1	7.95	0	7.95	Amazon
Load Sensor Modules	1	6.99	0	6.99	Amazon
Inductive Sensor	8	11.99	95.92	95.92	Amazon
Capacitive Sensor	12	9.99	119.88	119.88	Amazon
Stepper Motor	1	13.99	13.99	13.99	Amazon
Motor Driver	1	10.8	10.8	10.8	Polulu
12V 2A Power Source	1	10.69	10.69	10.69	Amazon
5V 4A Power Source	1	12.5	12.5	12.5	Amazon
Raspberry Pi v2 Camera	1	24.96	24.96	24.96	Amazon
RPi Camera Flex Cable	1	6.99	6.99	6.99	Amazon
GT2 Pulley Gear	2	8.99	17.98	17.98	Amazon
Timing Belt	1	11.99	11.99	0	Personal
Linear Rods	2	15	30	0	Personal
Linear Ball Bearing	4	11.89	11.89	0	Personal
LED Strip	1	14.99	14.99	14.99	Amazon
Trash bins	1	21.5	21.5	21.5	Amazon
Limit Switches	2	0.5	1	1	TechSpark
3D printed parts			40.5	40.5	TechSpark
Plywood			14	14	TechSpark
Wood for exterior			55.38	55.38	Home Depot
AWS			0	0.13	
		Total	514.96	498.02	

Figure 19: Budget