The Recyclinator: Automated Recyclables Sorting

Brandon Cheng, *ECE,* Tommy Chu, *ECE,* Nelusha Dias, *ECE/CS,* Cecilia Diaz, *MAE,*
Ryan Lin, *ECE,* Michael Mogilevsky, *ECE,* Simran Sareen, *ECE/CS,* and Hazem Zaky, *ECE*

***Abstract*—This project focuses on making improvements for the recycling process, by way of increasing its robustness and overall simplifying the process. Currently recycled materials are more expensive than new materials, by decreasing the cost of recycled materials, they become a more viable manufacturing option. Social, prototype, and experimental elements were considered for this project. The results include a working prototype that was able to sort each of the four materials appropriately, achieving the aforementioned positive social impact.**

Contents

I. Introduction

THE Recyclinator is an automated system that sorts through recyclable materials by material composition. A viable prototype was created that could be used on a large scale to aid in automating the recycling process. This was done using two sensors to sort between four main types of recycling: cardboard, plastic, metal, and glass. Current methods for sorting recycling employ multiple steps in the process to sort through the different materials, thus introducing multiple areas of failure and increasing the possibility for mistakes to be made. This project does all of the sorting in one step, increasing the overall system efficiency and reducing the number of possible failure points. Our main motivation for this project was first and foremost the environmental impact that will be discussed in the Social Element section, although it was also a passion project that allowed us to demonstrate how engineering can be used as an application to solve real world problems.

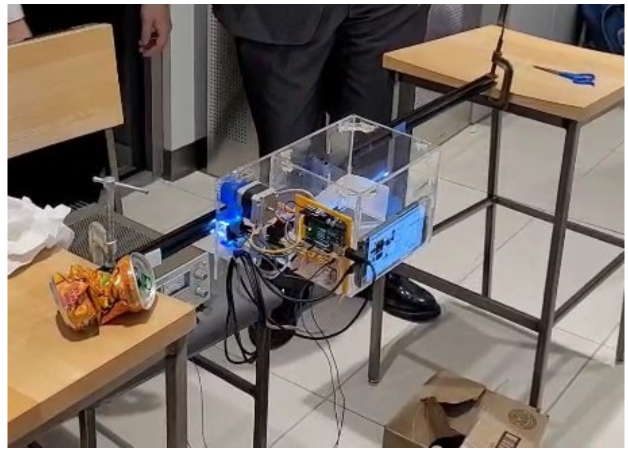

Fig. 1. Prototype of The Recyclinator

A. Social Element

The main social impact of our product comes from simplifying the sorting process for recycled material. Typical sorting methods are complicated, and are carried out in multiple steps, but our system accomplishes all of its sorting in just one. This reduces the possible points of failure in the recycling process. Additionally if our prototype is scaled appropriately it can help drive down the costs of producing new materials from recycled products. This would encourage businesses to make products out of recycled materials rather than producing new materials. Reducing the production of new materials will help reduce the amount of trash that inevitably gets sent to landfills further down the line.

B. Prototype Element

Our end product is a box that holds various electronics and sensors (Fig. 1). This box slides onto a rail using its wheel plate and is able to move itself along the rail with using a stepper motor. This enables the box to move back and forth across where multiple bins would be stationed. The bottom of the box is a trapdoor mechanism, which displaces the whole floor of the box to drop whatever item is placed inside. This enables us to move the item from within the box to the respective bin that the box is stationed over.

C. Experimental Element

One of our biggest concerns was making sure the capacitive sensor worked correctly, and that we understood how to use it. As such, before assembling our prototype we tested it against the different materials to make sure our equipment would work correctly. As mentioned previously initially when

Voltage	Material			
	Cardboard	Plastic	Metal	Glass
4	0	0	10	5
5	0	0	10	9
6	0	2	10	10
7	0	4	10	10

Fig. 2. How often the capacitive sensor was triggered for each material out of 10 trials.

	Material			
	Cardboard	Plastic	Metal	Glass
# successful trials / 20 trials	17	17	19	15
Accuracy	0.85	0.85	0.95	0.75

Fig. 3. From a total of 80 trials (20 trials for each material), the overall accuracy of the prototype is 85%

testing cardboard, plastic, and metal we saw the expected outputs, but had some inconsistencies when sensing glass. We tested our potentiometer at different voltage levels to see which one worked best for our purposes. The chart in fig. 2 shows the results of testing different voltage levels that the Arduino delivered to the capacitive sensor. As a result from our experimentation we determined that 5V was the optimal setting for the capacitive sensor, and decided to check each material 50 times so that if the sensor is triggered at all the material is identified as capacitive.

After the prototype was finished we mounted our system to two tables, and tested the four different materials to ensure that our whole system works, both the software, hardware, and overall design. We conducted our tests over multiple days to ensure that any errors from transporting our system were taken into account. We conducted 20 trials for each material type, and the chart in fig. 3 shows the accuracy of these trials. By examining how the accuracy changed for each material, as well as qualitative data we collected during the trials we were able to determine that the primary cause of failure was the capacitive sensor, and the least common was the color sensor since in only one case it failed to detect where on the rail the carriage was (that was the one failure during the metal trials).

II. Equipment and Methodology

A. Holding requirements

In order to be able to hold the desired item to sort and the electronic components, we needed to make a large and sturdy enough container. We used Solidworks to design a box, taking into account where we would attach our electronics and how they would interface with trapdoor as a moving part (Fig. 4). The size specifications of the box were based off of the average size of a water bottle, a commonly recycled material. At first, we didn't have any doubts on the structural integrity of PLA if we 3D printed the whole piece, but based on our time restraints and material availability, we settled on using acrylic to form the walls and floor of the box, due to it. Courtesy of

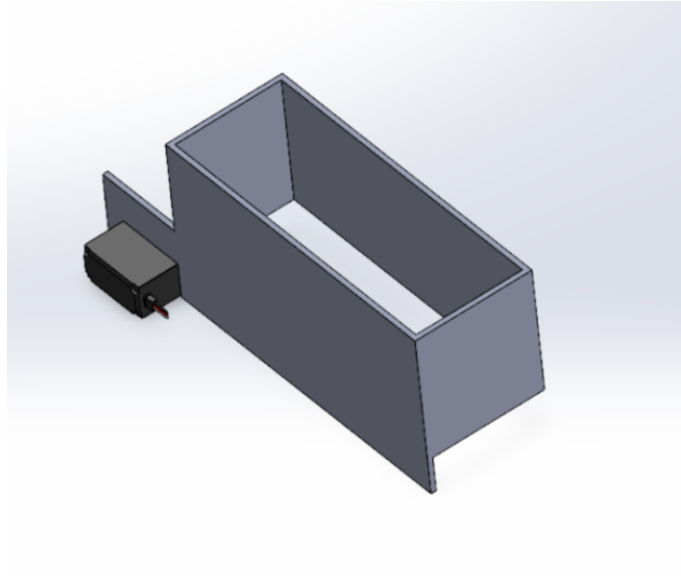

Fig. 4. Original CAD design of carriage system

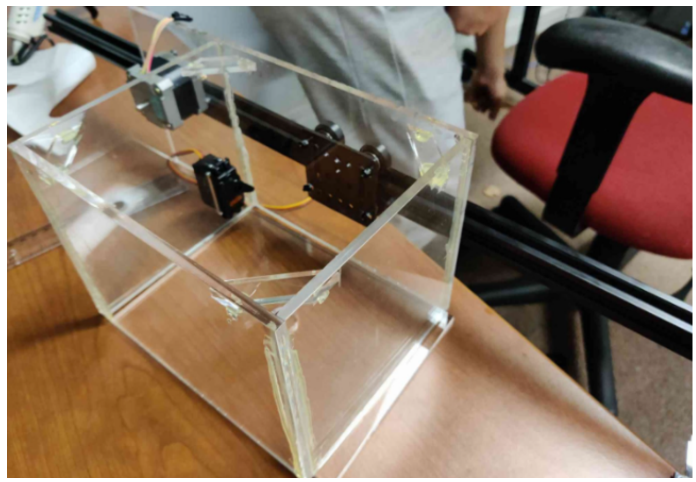

Fig. 5. Final carriage system design after construction with reinforced corners

the Rutgers Makerspace, we were able to precisely laser cut a sheet of acrylic to form the walls and trapdoor of the container. In addition to the dimensions of the wall and floor sheets, we cut holes in the walls to be able to securely attach various components to the box. For the walls, holes for bolts were cut to attach the wheel plate, the stepper motor, and the servo, while for the floor, a hole was cut for the capacitive sensor to fit through. The walls were then glued together using hot glue, which proved to be surprisingly strong. One miscalculation we made when putting the walls together was that the glue wasn't necessarily rigid enough to prevent the walls from parallelograming, so extra support triangles were fashioned to prevent this and maintain mechanical rigidity (Fig. 5).

For the trapdoor mechanism to drop the item, three extra pieces of acrylic were fashioned to translate the rotation of the servo into rotation of the trapdoor. We quickly found out that our initial design for a lever arm was faulty, but after a bit of tinkering, we finally settled on an appropriate set of lengths for the lever arms. A short acrylic bar would be hot glued to the horn of the servo to extend the lever arm of the servo, a second longer acrylic bar would be jointed to this first lever arm through a bolt and nut to make up the distance between the lever arm and the open trapdoor, and a final third mounting piece would be glued to the trapdoor itself, with a bolt and nut joint connecting the trapdoor to the long acrylic bar. In this configuration, we were able to open and close the trapdoor through a 130 degree angle rotation. Before we attached this lever arm, however, we made sure to fix the trapdoor to the back of the box through three hinges and set the rotation of the servo to 180 degrees so that the servo can be set properly. Another thing we found out when running the servo was that we had to slow down its rotation cycle so that it wouldn't draw such an excessive amount of current to the point that it would cause the Arduino to reset, since the servo was running on its 5 V output. This was prevented after implementing a simple software fix. In the end, we were elated that the servo was strong enough to be able to open and close the trapdoor with no issues at all.

B. Transportation requirements

In order to move our container between the different bins for each material, we settled on a rail system that our container can drive on (Fig. 6). A wheel plate holds the container onto the rail and allows it to roll on the rail freely while a wheel is pressed up on top of the rail to allow it to roll along the rail as we run the stepper motor. Although this system seems less reliable than a belt system in regards to the motor being attached to the box itself, which increases the weight of the box, as well as the possibility of the wheel slipping on the rail, because everything is contained on the box, it makes it easier for our system to be expanded in case we want to make the rail longer and add more categories to sort between. In order to mitigate the issue of the wheel slipping, we did three things. We adjusted the wheel plate so that the wheel would be comfortably in contact with the rail, which was helped by the weight of the box, we cast the wheel out of silicone, which would increase the coefficient of friction between the metal rail, and we built brackets around where the wheel contacts the rail to fix it in place and make it impossible for the wheel to slip off or fall off of the stepper motor.

Onto the stepper motor itself, this was bolted in a way such that the shaft would stick through a wall of the box and be able to reach the rail. The stepper motor is connected to a stepper motor driver on the breadboard so that we can control it through the Arduino. Luckily when we ordered the stepper motor drivers, heat sinks were also included to prevent the chip itself from burning, sparing us of any motor driver casualties. This stepper motor driver had to not only be connected to the stepper motor and the Arduino, but also supply of 12V for the stepper motor itself to run. Although it is not at all elegant, having an external power supply connected to the board through two long wires sufficed in providing enough power to the motors and enough length for the amount of distance we were traveling along the rail (Fig. 7). Possible solutions to fix this issue are addressed in the future work and improvements sections in case of rail expansion.

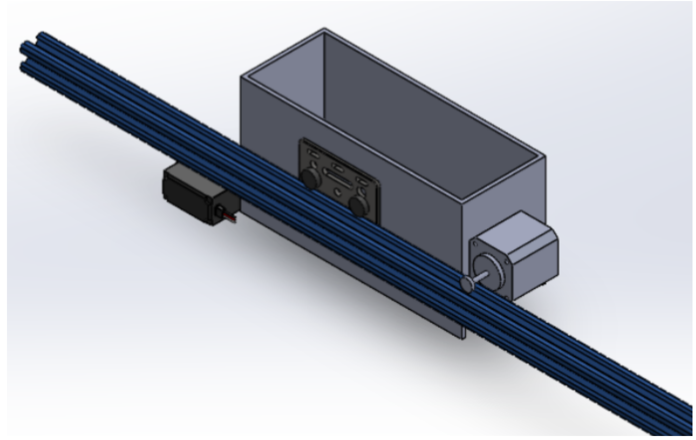

Fig. 6. CAD design for with rail and stepper motor

Fig. 7. Power Supply and Stepper Motor

To keep track of where the container is on the rail, we used a color sensor. The sensor looks through the clear acrylic wall to sense tape demarcations on the rail of different colors. Since the rail itself is black, we used blue tape to mark out three sections of the rail for the three materials than we want to stop in case we move, as well a white demarcation at the beginning, which doubles as the fourth material that we sort out, in the case that we don't move on the rail, and as a carriage return marker when we want to reset after moving on the rail. Although we would have been able to use different colors for each of the four sections with the flexibility of the color sensor, using a counting variable to increment when we detect specifically blue tape would be easier and cheaper in terms of only having to deal with three colors (white, black, and blue), be easier to scale past the standard red-green-blue plus cyan-magenta-yellow combinations, and overall be more reliable in terms of color differentiation. Furthermore, it made it easier to map where we want to go based on the variables

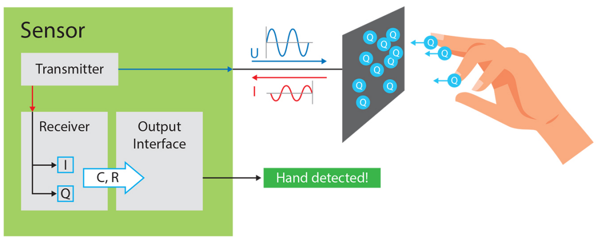

Fig. 8. Graphic showing how a capacitive sensor works, courtesy of ams [1]

we are given with the counting variable. For example, if the capacitive sensor return true, we would add 2 to the position, while if light was detected, we would add 1 to the position. This gives us our four combinations of positions 0+0, 0+1, 2+0, and 2+1 that we encode to the position of the rail.

C. Sensing requirements

1) Capacitive Sensor: In order to distinguish between metal and glass from other materials, we used a capacitive sensor. When these materials come close to the end of the capacitive sensor, it disrupts its generated electric field, giving us a non-zero reading from its signal wire (Fig. 8). This was a new component that none of us had worked with before, so we did some testing with the sensor at different input voltages as well as how tuning its on-board potentiometer affected its reading. We found that high input voltages allowed for the sensor to sense materials at slightly longer ranges and trigger for a wider range of materials while it's potentiometer affected how sensitively the sensor was for a particular material. What was particularly useful about this sensor was the built-in red LED at the bottom, which helped us diagnose whether or not the capacitive sensor was detecting anything at all. Through our testing, putting in 5V from the Arduino seemed to give us the desired level of differentiation, although it was a bit unreliable with glass. With glass, we saw that the on-board LED would flash on and off sporadically. In order to stabilize reading, when the Arduino would sense from the capacitive sensor, it would take 50 samples and increment a counting variable whenever a non-zero value was detected from the sensor. If the counting variable increments at all, then we say that the material is capacitive. This greatly decreased the number of false negative readings for glass. Another note with the capacitive sensor was that while our unit was only able to output binary values, our research indicated that there are capacitive sensors that are able to output an analog signal based on the material, which might be able to remove the need of the light sensor discussed in the next section.

2) Light Sensor: In order to distinguish between glass and plastic from other materials, we programmed a light detection algorithm that would detect how much light passes through the object. The light source came in the form of a bright LED, which we put in front of a paper towel in order to diffuse the light, while the light sensor itself was in the form of a phone camera, which we mounted to the front of the box. To detect both when objects are placed in the box, as well as how much light passes through them, we created an Android application in Flutter. This application displays a view of the phone's camera, which can have a box drawn on it to select the region that is observed by the algorithm.

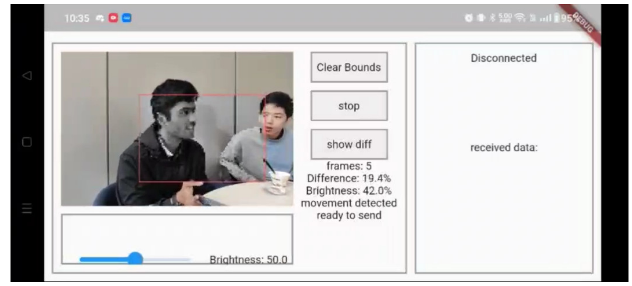

Fig. 9. Phone App: Gray-scale Conversion

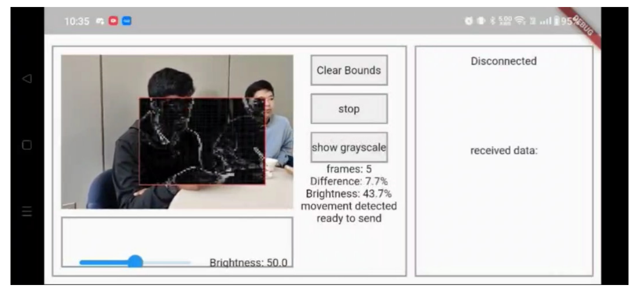

Fig. 10. Phone App: Gray-scale Difference

To detect when an object is placed within the box, the image data within the selected region is first converted to gray-scale (Fig. 9). Then, every 250 ms, a snapshot of the gray-scale region is stored, and compared with previous snapshots. The percentage of pixels that changed between the previous and current snapshot is visible on the screen (Fig. 10). An object is detected when the percent change crosses an upper threshold, which can be set through a slider on the screen. Then, once the percent change crosses a separate lower threshold, it is determined that the object has been placed and is no longer moving, and the program can move to the next step.

After determining that an object has been placed, the program calculates the brightness of the region as a percentage of the maximum possible brightness by adding the gray-scale value for each pixel, and dividing by 255 times the number of pixels. If this brightness is above a threshold set by a slider, then it will send the Arduino a "1", signalling that light passes through the material. Otherwise, it will send a "0". The phone then waits for the Arduino to report that it is finished with its operation, and the program will start waiting for the next object. To tune the system, we first place the phone in the mount on the front of the box and selected an appropriate region that includes the area lit by the led. We then used the display of percent change and brightness and observed how much they changed when placing various objects, and lastly set the thresholds through the sliders available on the screen.

III. Results and Discussion

In terms of the material detection system, the material is correctly identified roughly 85% of the time. Although there are only 4 different categories of materials in which the product can detect, there are many different types of each material that can be detected.

Most of the error has occurred because of the capacitive sensor. The capacitive sensor is extremely sensitive when attempting to detect glass. For example, giving the capacitive sensor 5V was not enough to get it to detect glass, but giving it 7V was. This shows that detecting glass was testing the limits of the capacitive sensor. When detecting metal, the capacitive sensor had no problems identifying whether the material disrupted the electric field emitted by the sensor. The light sensor had a much lower chance of producing an error in correctly identifying the type of material being recycled. Since the items being recycled mainly allowed light to pass through or absorbed light, the camera sensor did not have much of a problem detecting the material. The color sensor that was used to detect where the container was supposed to stop had almost 100% accuracy. This is because the color sensor was able to accurately detect the changes in color on the rail so that it knows exactly where it is as well as how far it is from its destination.

A. *Future work and improvements*

Although our transportation system was able to successfully transport the recyclable material to its destination, there are still a few improvements that could be made. When the container was being transported across the rail, the wire from the 12V power supply was following it the whole time. Although this may have been acceptable for only having 4 different types of materials being detected, this would be impractical if we had more types of materials being detected. If the number of materials being detected were to hit a larger number, for example 20 materials, the wire following the container would simply be too long and cause multiple hazards. A possible improvement for this would be to connect a battery on board the container. This would eliminate the need for the outside power supply with wires following the container. In terms of charging these batteries, a wireless charging pad could be set up at the original position of the container so that the battery charges whenever the container is not in use. In regards to the color sensor, it is possible that the color sensor could miss the mark of where it is supposed to dispense the recyclable material. An improvement would be to replace it with a distance sensor, allowing the mechanism to know its location based on how far it is from the starting position. Doing this would also eliminate the need for tape on the rail, allowing for the transportation system to smoothly move across the rail.

The container used for transportation was able to successfully hold the materials being recycled until they were dispensed in their respective locations. However, the container could be improved during the detection phase of the program. When a material is being detected, it has to be placed directly on the capacitive sensor, which is protruding from the bottom of the box. Because of this, one has to balance the material on the capacitive sensor so that it could accurately detect whether the material disrupted the magnetic field. An improvement to increase the reliability of the capacitive sensor would be to increase the input voltage to increase the range of the sensor, add more sensors, and have the capacitive sensors sit almost flush with the rest of the container. However, it should be noted that the capacitive sensor should protrude from the base of the container at least a few millimeters so that it can make contact with the material being detected.

The light sensor used to detect the transparency of the recycled material was implemented through an app on a phone. However, if we switched it so that the algorithm was hosted by a Raspberry Pi, the error in detecting materials would decrease. This is because the Raspberry Pi is able to run the algorithm as well as control the hardware components of the system. Using a Raspberry Pi would increase the reliability and the range of the capacitive sensor, allowing it to be much more accurate than it currently is. As stated earlier, the two sensors were able to correctly identify the material roughly 80-85% of the time. This could be improved using a machine learning algorithm for the camera. Using a machine learning algorithm would allow for the camera to continuously improve its accuracy when detecting different materials such as glass and cardboard. Another issue that the light sensor continuously ran into was that it couldn't accurately detect materials that were colored plastic or opaque. This part could also be fixed through the use of machine learning, as it would help identify whether the material was plastic through the use of previous data.

Moreover, machine learning would help expand both the breadth and depth of the categories being detected. Due to the fact that the current algorithm only tests whether light passes through and whether the material is capacitive, we are naturally constrained to a total of four broad categories (metal, plastic, glass, and cardboard). When it is desired to recycle a product made of multiple material categories (e.g., a product made of both metal and plastic), our algorithm will only detect it as one category; since mixed-material items are difficult to recycle due to the requirement of properly isolating the materials for reusability, the current state of our project is severely limited. Additionally, simply extracting data from light-sensing and capacitive-sensing allows for a detection of any metal regardless of its type; in reality, detecting diverse types of metals and other materials is important (e.g., alloys need to be carefully separated because some metals may release toxins into the environment), which further reduces the practical capabilities of the current prototype.

Through the use of machine learning, additional parameters can be naturally integrated with the current material-detection algorithm. In a Convolutional Neural Network, tens or hundreds of unique layers are used to detect different features on an input image by applying a convolving filter to each training image with different resolutions, and then using the result as an input to the next layer. Although simply using a Convolutional Neural Network may provide enough practicality in some cases, we can further improve it by introducing a residual connection, which would would expand on the functionality by pipelining a connection between the output of one convolution layer to another several layers ahead.

This allows for a construction of networks with thousands of convolutional layers, which enables us to add more layers to the residual training blocks to expand the breadth of materials detected without affecting training error or accuracy/precision during the mapping of results to their respective categories.

Additionally, fine-tuning a Convolutional Neural Network with pre-existing datasets (e.g., TrashNet has over 2,500+ images of various recyclable materials such as glass, paper, cardboard, plastic, metal, and trash) can be easily achieved due to our project already integrating a phone camera for light-sensing. Therefore, amalgamating capacitive-sensing, light-sensing, image-detection, and other parameters (such as weight of the material, organic compounds, etc.) will allow for more specific categorizations.

IV. Conclusions

Overall, the purpose of our project was to design a mechanism to sort recycling in order to make the process more feasible for daily use. In order to make the prototype we focused on the social, physical and experimental elements. The social element provided the foundation of our project. The current methods available for sorting require several steps and have a high chance of failure, and oftentimes people do not complete their end of recycling either. The negative impact of these human and machine errors motivated us to design a prototype that required less steps and also did not require any physical sorting on the users end. From here we branched out to the physical element where we actually designed our model. With the use of store bought material and 3D printed material we built the physical design. Our electronic components, the Arduino, stepper motor, servos, capacitive sensor, color sensor, and light sensor, provided the mobility and detection system we needed in order to physically sort out the material. Looking forward we would like to make adjustments to the design so that we can sort through objects faster, and make adjustments to the hardware and software to detect objects more efficiently.

Acknowledgements

We would like to thank Dean Antoine, our design advisor Jack Piro, our alumni mentor Joshua Posen, and the rest of the design advisor team for making this project possible, as well as for their guidance throughout the semester.

References

[1] *Capacitive Sensing*, Ams, https://ams.com/capacitive-sensing