Product: Recyclotron
Team: Group Eight

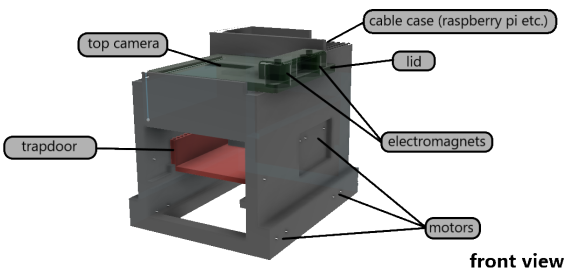
Abstract

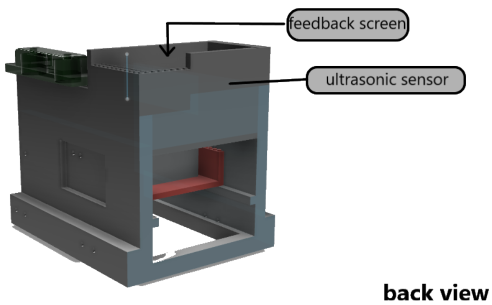
This past sprint focused on increased functionality, such as implementing a trigger to detect when the user puts trash in and detecting unseen objects better. We also started extending the system, allowing for more accurate sorting into 4 bins and ultrasonic sensors to detect which bin Recyclotron is over.

Goals for this sprint:

- Add a lid and chamber above trapdoor to hold rubbish, cameras and space for wires
- Add a trigger to detect rubbish and start camera detection
- Increase structural stability and scale by 3D printing parts
- Improve ML Model to have low false positives
- Allow the ML model to work with 3 and 4 categories, not just 2
- Show clear feedback of what category rubbish is on screen

System Model (spaces indicate where components will go):





1. Project Plan Update

We aimed to show a fully integrated and tested interactive Recyclotron. Unfortunately, we had a lot of setbacks while working towards this demo, which hindered our progress. They are detailed in this report. Even our Raspberry Pi's SD card was corrupted with no back up, meaning we had to re-code it. In future we will back up code daily and use version control in GitHub to minimise this risk. Despite being unable to extensively test the 3D printed model in time for the report, we completed our goals, testing and integrating instead with the Lego model.

1.1. Software Team

Zhixing set several tasks to improve the ML model:

1. Collect images for own dataset, create server to store, tag and use them as test data
2. Classify input based on the number of bins
3. Add and test confidence intervals
4. Fine tune the model to penalise false positives

1.2. Hardware Team

Sakib split hardware tasks into two sub-teams, modelling and components.

Modelling Team:

1. Model the current system in modelling software
2. 3D print current system, attach motors and test model still works
3. Scale up the design and reprint (for a larger tray/platform)
4. Model and print lid, construct and test mechanism

Components Team:

1. Allow the car to know where it is, integrate with raspberry pi, testing car stops and trapdoor/slider opens
2. Implement a screen display to show item's category
3. Create a lid trigger, integrate with raspberry pi, test closing lid triggers cameras, test lid locks when moving

1.3. Group Organisation

The code for the Raspberry Pi, machine learning and the server is stored on GitHub. Autodesk was used by modelling team to share models, and Fusion 360 was used to create them. Overleaf was used to create the report, and Google Slides to create the presentation.

We decided to change how progress is monitored. Instead of using Trello we now use a daily log to track progress. The daily log is stored on Google Docs and can be edited by any team member once they have completed a task.

Often new problems are encountered and new sub-tasks are created, which means tasks have to be updated frequently. Sometimes Trello is not updated because a task is not fully completely on that day, even though it was worked on. Conversely, in a daily log, progress is clear as everyone records what they've done each day. Team members can also be much more specific, commenting on difficulty. This allows the team to see who has done which tasks.

1.4. Future Milestones

Our milestones have been updated to reflect progress. In particular we removed the functionality of crushing. We feel this would be too difficult to accomplish and is not necessary to fulfill our user needs. The others are now more specific.

Milestones	Purpose
Demo 3 - Increased Modularity	Reduces amount of sorting with machinery and people councils do
Demo 3 - Better feedback and user interaction	Lights helpfully indicate full bins, User input improves accuracy and training
Demo 4 - Larger Tray and fully 3D printed model	Allow disposal of larger objects, currently cannot fit wide or tall objects

2. Technical Details

2.1. Feedback Screen

In order to provide clear feedback we are using a Raspberry Pi screen. This visual feedback reassures the user that the rubbish has been disposed of. Currently the category of trash and bin into which it is dropped into is displayed. In future the user will be able to use a touch screen to manually select a category, and will also alerted if a bin is full.

2.2. Electromagnetic Lid

We needed a trigger system to detect when rubbish has been put into Recylotron. We decided to use a lid system, where the user opens a lid to put rubbish in, and once the lid closes, it locks and trash detection begins.

In order to tell us when a lid is open or closed, we use a magnet (separate from the electromagnets) that completes a circuit using a reed switch.
The shutting of the lid triggers the cameras, which then take pictures of the trash. The electromagnets receive a voltage signal activating them to attach to a piece of metal, effectively locking the lid. The lid is unlocked once the trash has been disposed of.

We had other ideas for a trigger to start camera detection. A sound sensor would use the noise of the rubbish being dropped in, a weight sensor would detect the increase in mass from the object, and an IR mechanism would detect disruption of a constant beam.
Sound sensors require calibration in order to work with both quiet and loud trash (think paper and metal), as well as discern from background noise. This means they would constantly collect data, consuming power excessively.
Likewise, weight sensors are difficult to calibrate (different trash objects have different masses) and constantly consume power.
We decided against an IR mechanism as a magnetic circuit is more reliable and simpler to implement than an array of IR sensors.

2.3. Ultrasonic Sensors

In the first demo the car had didn't know where it was in relation to the bins - all movement was hard-coded and constant. However, for Recyclotron to be extendable (more bins can be added freely) we need to track the cars location.

One idea used lasers, a reflective surface and a sensor but this was too complicated, with too many angles and scenarios to test. Furthermore we would need to add extra sensors, so it'd be difficult to add more bins.

Instead, we're using two ultrasonic sensors, one firing vertically below and another firing horizontally across the rails. The car starts at a known location, and once it starts moving the horizontal ultrasonic sensor is used to detect if the car is in the correct area. The vertical sensor will be used in future demos (detect if bins are full).

We decided to use ultrasonic sensors as they're available with the GrovePi, detect at the range we need (3.5m) and their accurate resolution is 1cm.

2.4. Camera Placement

We decided to place cameras on the top and side of the chamber. Having two different views increases the chance that we can see the objects most distinctive features. Together, the two cameras together provide ample information for classification. A bottom view of the object would also be difficult to achieve as a camera could not be placed on the trap door.

We have not tested other camera locations yet, as the current Lego chamber is multicolored (confusing the object recognition), and the 3D printed chamber has been postponed.

2.5. Metrics for Sorting

We improved our model to recognise a maximum of 4 categories, extending beyond just 2 bins. This allows clients to recycle waste into fine-grained categories.

The model's categories are adjusted depending on the number of bins:

2 Bins - Non-recyclable and Mixed Recyclable (metal, card-

board, plastic, paper, glass)
3 Bins - Non-recyclable, Metal and Mixed Recyclable (cardboard, plastic, paper, glass)
4 Bins - Non-recyclable, Metal, Cardboard and Mixed Recyclable (plastic, paper, glass)

In choosing categories, we didn't separate glass and plastic because they're hard to distinguish using vision. Paper is also difficult to distinguish, especially if it is flat. The other categories are ones used in recycling centres.
Although we use a new metric for each differing quantity of bins, our overall model is still the same.
Primarily we want to minimise false positives - in particular, we don't want trash being misidentified as recyclable.

According to UK recycling authorities, mixed-recycling contaminated with non-recyclables is incinerated instead of being recycled. [1]
One council remarked, "We are charged extra if the recyclable material is contaminated with other materials as it slows down the recycling process and can cause damage to the machinery" [2]. Therefore, we penalise false positives to avoid trash contaminating recycling bins.

On the other hand, if a metal object contaminates the cardboard bin it is not as problematic. This is because local recycling centres sort for metal and cardboard anyway.[3]

While we want to reduce the number of trash items contaminating recyclable bins, we can't solely use that as our test metric. We could easily achieve having no misclassified trash in recyclables, by putting everything in trash! We still value overall accuracy, but now consider recyclable accuracy and trash recall, which reduces the rate of false positives. Test results are detailed in a confusion matrix in section 3.3.

2.6. In-Domain Data

In the last demo we tested the model using 4 online datasets, instead of testing on images taken within Recyclotron.
As we were still constructing the chamber for this demo, we couldn't test using images directly from the chamber. Despite this, we collected our own dataset of trash (around 500 images we took ourselves) allowing us to test for specific items, and have more representative tests.
The images are uploaded to a server for storage, where they're scaled to the resolution of the webcam photos.

2.7. Old Model Problems

Before the first demo our test results showed high accuracy for object recognition. For example, our KNN model (trained on neural representations extracted from ImageNet-pretrained resnets) had 89% recyclable accuracy.
However, when we tested the same model on our own dataset, we saw a huge drop of performance - from around 90% accuracy (70% overall) to 40%.

This drop is due a problem in our initial dataset, which was compiled from 4 different sources.
We created a balanced set of recyclable and non-recyclable items, with recyclable items mostly coming from one single dataset called TrashNet. The images were then split into the training or the test set, and once trained we evaluated accuracy using the test set.

However, we found that TrashNet tends to have multiple photos of the same item, resulting in a large overlap between the training and test set. The model had essentially "seen" many of the test items before. Thus our first tests don't completely portray performance on new or unseen items.
We fixed this issue by ensuring we trained and tested on completely different datasets - training with an online dataset, testing with own dataset. Yet we found that our model still couldn't generalize to new items.

2.8. Model Improvements

Hence we altered our initial model, collected more data, and trained an end-to-end neural net. The neural network output a class prediction directly from an image, without a classifier like KNN.
We used Litterati, a global website where people upload photos of trash with nearly 5 million user-tagged trash images, as our training dataset. [4]
First, we cleaned the data so it matched our webcam input, by ensuring all images were of dimension 512x512 and JPEG compression quality of 70-80%.

We then re-labeled the images by defining a mapping between the Litterati user tags to five categories we chose (glass, metal, paper/cardboard, recyclable plastic and trash). Finally, we downsampled the trash class to match the number of recyclable items.

2.9. New Model

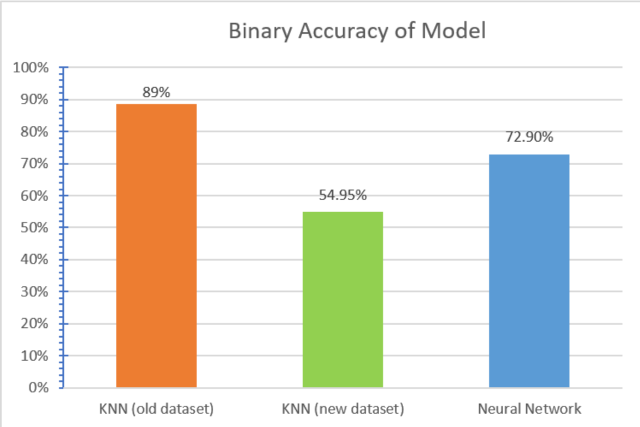
For this demo we trained a neural net initialized with ImageNet weights. Although our model is adapted from the ResNet50 model as before, it has changed.
Features from different levels of abstraction are combined together, and the final layer uses not only high level class information (such as whether the item is bottle or can), but also more low level details like texture, reflection and transparency.

3. Evaluation

3.1. Old vs New Model

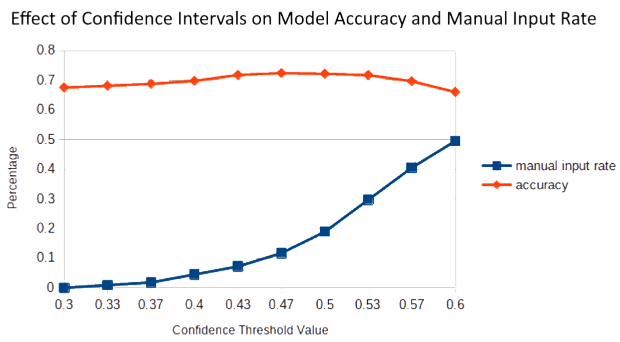
After discovering our initial dataset was faulty (KNN's 89% accuracy is due to overlap in training and test datasets), we fixed and re-tested our models for recognition of both recyclable and non-recyclable trash.

As seen below, KNN was around around 55% accurate (using Literrati to train and our own dataset to test), whereas our new Neural Network model achieved around 70% accuracy. This is a considerable improvement.



3.2. Confidence Thresholds

We've also tested whether confidence thresholds help, by asking the user to manually categorise the trash when the system is unsure about its prediction. If the maximum prediction score is below the threshold the user will be prompted for input.



We can see that the optimal confidence threshold value of around 0.47 improves overall accuracy by 5%. In exchange, it asks for user input about 10% of the time.
Raising the threshold further won't help because often the model is too confident its prediction is correct. Asking the user repeatedly for input would also defeat the goal of automatically sorting rubbish.

3.3. Recall and Accuracy between Categories

Here is the confusion matrix obtained for our new model (neural network trained on Literratti, tested on our own dataset).
Each column shows where items in that column were categorised. Each row shows where items in that category came from. A category's recall is how many items that belong in the category are classed in that category. For example, plastic has recall of 48% because roughly 48% of plastic items are put elsewhere, instead of being identified as plastic A category's accuracy is how many items belong in the category they were identified in. For example, metal has a 67% accuracy as 33% of items identified as metal belong elsewhere (in trash for these tests).

We see most errors are between plastic/paper and trash,

	Glass	Metal	Paper	Plastic	Trash	Accuracy
Glass	0.0	0.0	1.0	0.0	0.0	0.0
Metal	0.0	2	0.0	0.0	1.0	0.67
Paper	0.0	0.0	20.0	1.0	10.0	0.65
Plastic	0.0	0.0	0.0	12.0	2.0	0.86
Trash	0.0	0.0	5.0	11.0	42.0	0.72
Recall	N/A	1.0	0.8	0.48	0.76	N/A

although we see that trash contaminates every category to a differing degree. Glass fails to be identified correctly with a recall of 0, being misidentified as plastic.

3.4. Modelling and Component Testing

Modelling took longer than expected, often sizes of parts were incorrect (e.g. tray teeth being too large for cogs). Designs needed to be finalised, tested within the software with measurements being double checked. Then the model was exported and printing began.

Additionally, the prints themselves take an extremely long time. The body took 3 days to print, and the chamber lid took 1 day to print. If there are any failures a print is completely restarted, wasting any material used. We've had to reprint the tray/sliding door 3 times due to sizing issues, reprint the lid of the chamber 3 times as the printer ran out of plastic and we did not have enough time to print the body even once. The rate of turn around was just too slow for us to test models once printed.

The LCD screen was tested 5 times to see if we can output the classification to the display. The LEDs were checked 5 times to see if they switch on with the trigger. The electromagnet trigger was tested lid 5 times ensuring that it sent the correct signal to raspberry pi. The ultrasonic sensor was tested 5 times to ensure it stopped the car. Every component operated successfully. We plan to more extensively test for the next demo (once components are integrated fully, we can test the system as a whole).

4. Budget

Below is a table showing our budget so far. These are estimates and subject to change. For this demo, we more accurately estimated the cost to our budget, even though we have not spent much (as the University has provided for us). In future demos we don't plan to spend more, as we identified all our components in this demo.

Component	Cost
3D Printed Body/Lid	£65
3 Electromagnets	£4
2 Ultrasonic Sensors and LCD Screen	£10
2 LEDs	£3
2 Webcams	£15
5 EV3 Motors	£130
2 Steel Railings	£10
Higher End Raspberry Pi	£50
Total Cost	£287
Total Spent	£97

References

[1] Royal Borough of Kensington and Chelsea, Why some items can't be recycled
https://www.rbkc.gov.uk/bins-and-recycling/rubbish-and-recycling/recycling-and-rubbish-faqs/why-some-items-cant-be-recycled

[2] Sandwell Metropolitan Borough Council, What happens to your recycling and rubbish?
http://www.sandwell.gov.uk/info/200160/bins_and_recycling/496/what_happens_to_your_recycling_and_rubbish

[3] The Waste and Resources Action Programme, Recycling Centres: How is it recycled?
https://www.recyclenow.com/cy/node/1911

[4] Literatti, Literatti Global Map
https://map.litterati.org/globalmap/