Team 15 Documentation
Auto-Sorting Trash Can

William Martino, Alexandra Michael, Casey Price, John Ventura

Table of Contents

Introduction

Our project derives primarily from the problem of inaccurate waste sorting when disposing of items. While most people know the basics of what sorts of waste is recyclable, compostable, or landfill material, subtler differences between items (e.g., recyclable clean paper vs. compostable dirty paper and napkins) are often missed. People also often simply don't have time to think about which materials belong in which category. The result is a loss of materials that could have been reused, but instead go to landfill, where they unnecessarily increase pollution and greenhouse gas emissions.

We designed our project to mitigate this problem by building a trash can capable of automatically sorting disposed items into the recycling, compost, or landfill categories. The project would enable people to avoid worrying about which items are recyclable and which are not, without contributing unnecessarily to pollution. The user could drop a waste item into the upper compartment of the bin, and the bin would automatically take a picture and determine the category using a neural network. One of three interior compartments would open depending on the result, dropping the item into the appropriate section and completing the sorting process.

Materials

Components

- Raspberry Pi
- Camera module for Raspberry Pi
- 3x Servos
- LED
- Jumper wires & breadboard
- Cardboard
- Duct tape

Software and Tools

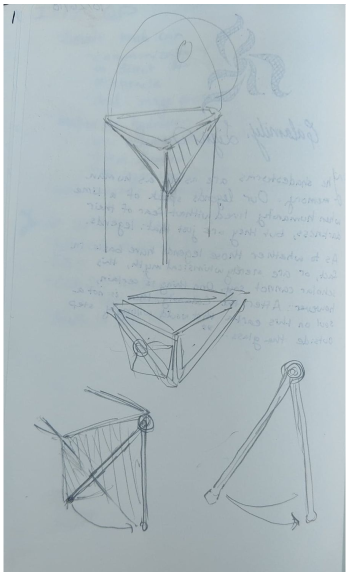
- TensorFlow and TensorF low Hub
- SolidWorks

Process

Designing the Trash Can

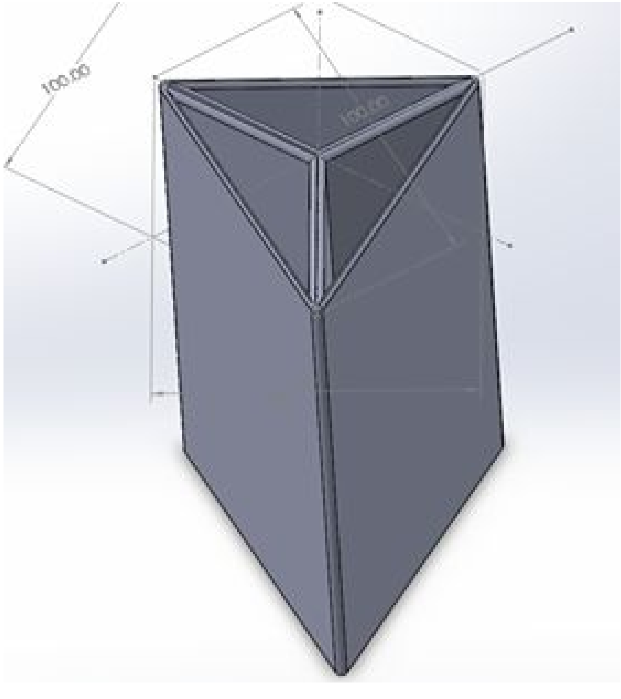
Our first step was to develop a sketch of how we wanted the trash bin to look and function. We settled on a design that involved four compartments: one primary one in the upper layer of the trash bin, where a waste item would initially land, and three more splitting the lower part of the trash bin into equal portions. In order to conditionally drop an item into one of the three lower compartments, we decided on stepper motor-controlled flaps that would drop down to let items in on command by the Raspberry Pi.

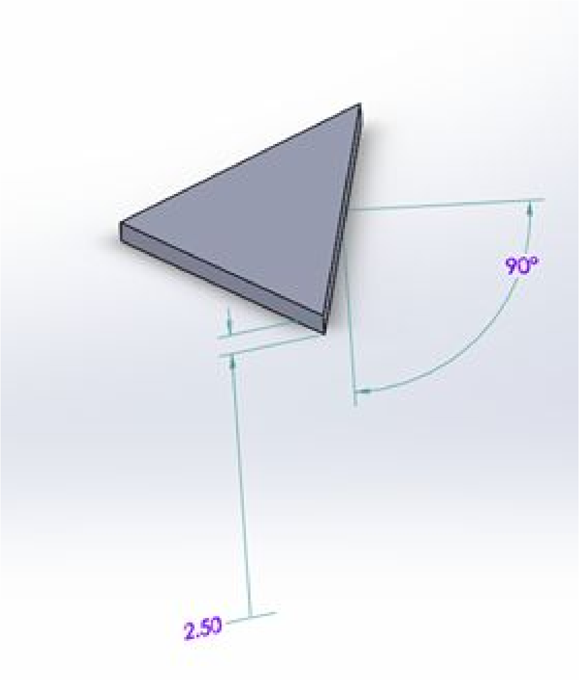
Our initial sketches:

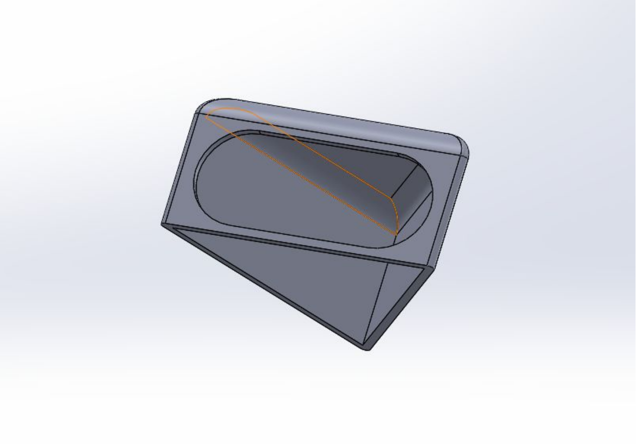


After developing the basic idea, we used Solidworks to build a full model of the prototype.

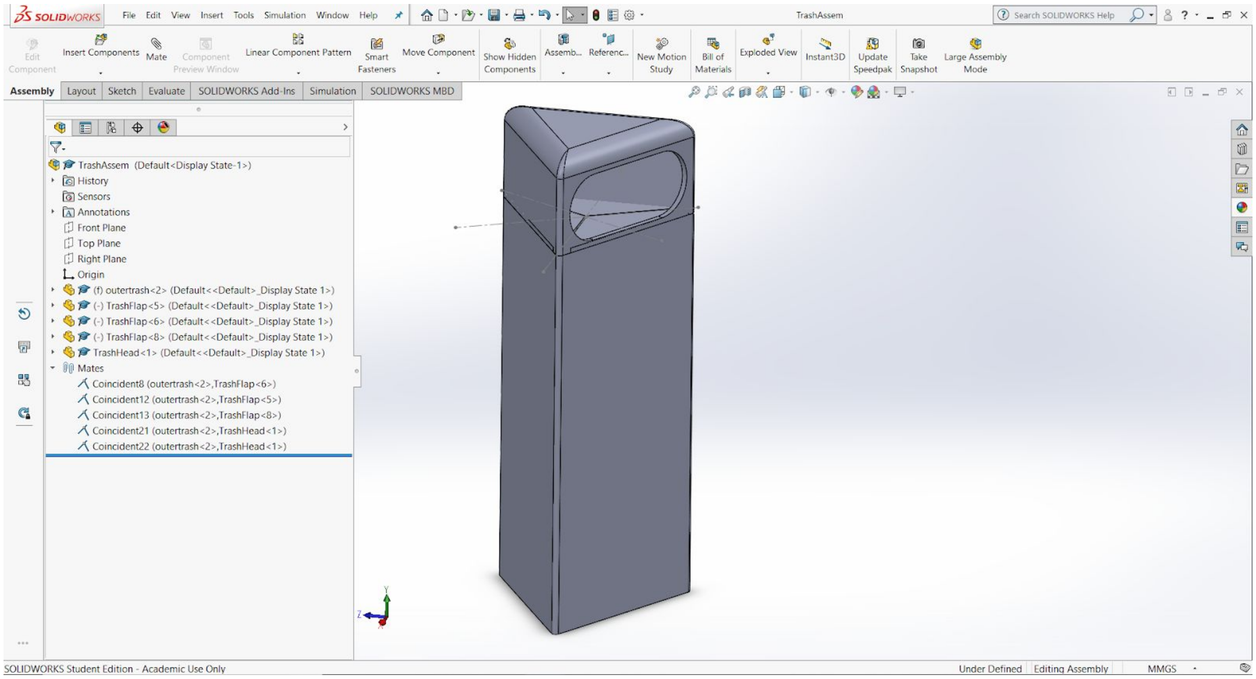
Components of our model:







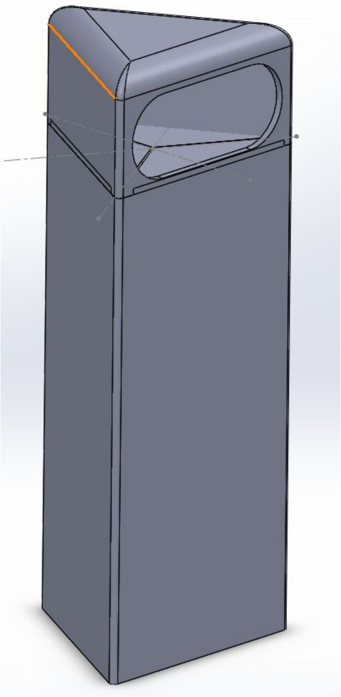
The finalized model:



TrashAssem (Default<Display State-1>)
- History
- Sensors
- Annotations
- Front Plane
- Top Plane
- Right Plane
- Origin
- (f) outertrash<2> (Default<<Default>_Display State 1>)
- (-) TrashFlap<5> (Default<<Default>_Display State 1>)
- (-) TrashFlap<6> (Default<<Default>_Display State 1>)
- (-) TrashFlap<8> (Default<<Default>_Display State 1>)
- TrashHead<1> (Default<<Default>_Display State 1>)
- Mates
  - Coincident8 (outertrash<2>,TrashFlap<6>)
  - Coincident12 (outertrash<2>,TrashFlap<5>)
  - Coincident13 (outertrash<2>,TrashFlap<8>)
  - Coincident21 (outertrash<2>,TrashHead<1>)
  - Coincident22 (outertrash<2>,TrashHead<1>)

SOLIDWORKS Student Edition - Academic Use Only

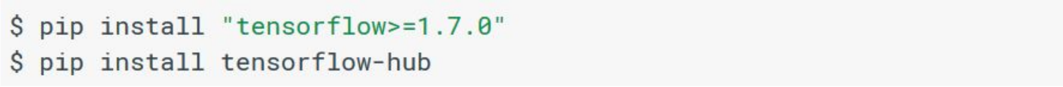
Under Defined | Editing Assembly | MMGS



Training the Neural Network

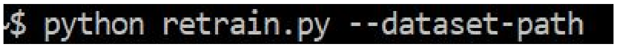
Simultaneously with designing the trash can, we used Tensorflow and its transfer learning tool, Tensorflow Hub, to train our image classification neural network by modifying select layers of a preexisting network. Our first step was to download and install Tensorflow, using the instructions found on the official website: https://www.tensorflow.org/install/pip.

We then installed Tensorflow Hub, Tensorflow's transfer learning tool, using the following commands:



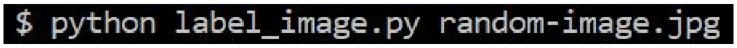
```
$ pip install "tensorflow>=1.7.0"
$ pip install tensorflow-hub
```

Once Tensorflow Hub was installed, we downloaded a trash-oriented data set from github at https://github.com/garythung/trashnet/blob/master/data/dataset-resized.zip and resorted the images in the dataset into compost, recycling, and trash (landfill) folders. We then used the retrain.py transfer learning program to retrain the provided network, passing in the labeled data directory as an argument:



```
$ python retrain.py --dataset-path
```

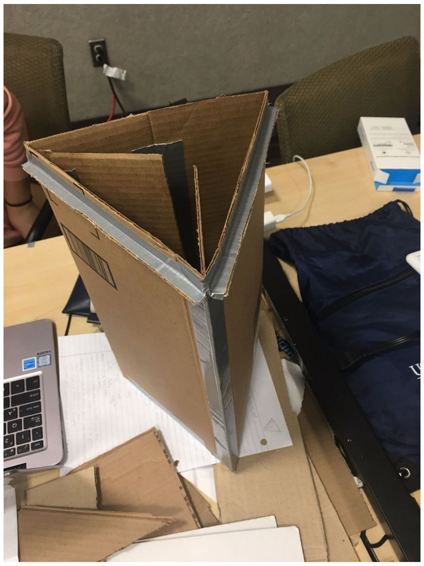
Once the retraining was complete, we tested the network on random images using the label_image.py program, and transferred the completed network to the Raspberry Pi.

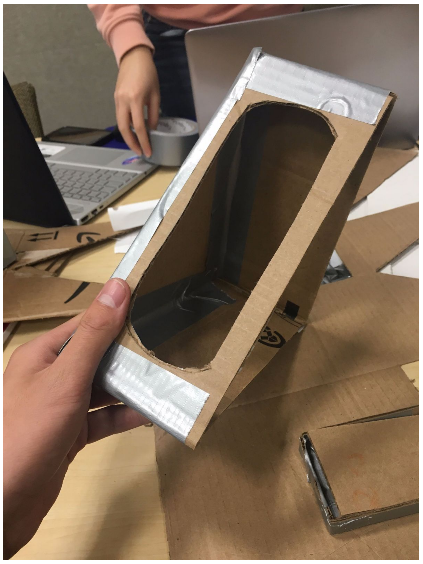


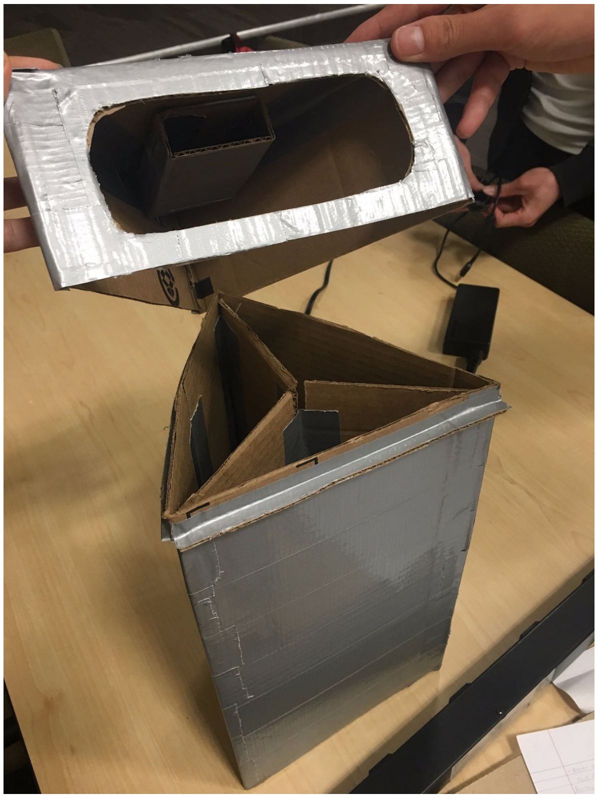
```
$ python label_image.py random-image.jpg
```

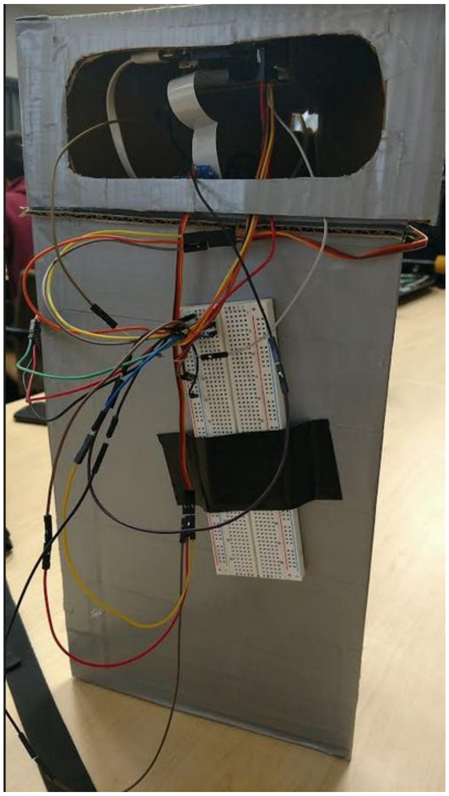
Assembling the Prototype

We used cardboard and duct tape to construct the physical prototype in imitation of our Solidworks model, as shown below. We used jumper wires and a breadboard to connect the Raspberry Pi, which was stashed in an internal compartment, to the power source, camera module, and an LED to facilitate pictures.









Conclusion

If we had more time, funding, and energy, we would have liked to further refine the design. As it stands, the bin would have trouble dealing with people dropping in multiple items of different categories at a time, and some items would end up in the wrong categories. We would have liked to work out a solution to this problem. Additionally, we definitely would have constructed the prototype using a more durable material, and would have used more precise measurements by either 3D printing the parts or by laser cutting them out. We would also make a larger prototype in order to accommodate larger samples of waste.

Overall, our greatest obstacle during the process of this project was assembling the final prototype, because we ran into unexpected interference between the flaps and the sides of the trash can when the flaps rotated. Our proudest moment was when we finally got all of the trash can flaps to turn correctly, because we had to make a lot of adjustments to allow them to work.

References

The SolidWorks help forum was very helpful with figuring out functions of designing the CAD model, such as mirroring cuts and rendering the final product.
http://help.solidworks.com/

The following sites provided data regarding the problem we were attempting to solve:
https://www.cleveland.com/metro/index.ssf/2018/06/recycling_were_doing_it_wrong.html

https://www.greenamerica.org/rethinking-recycling/americans-are-really-bad-recycling-only-because-were-not-trying-very-hard

We used the following resources to assemble and test the neural network:

https://github.com/garythung/trashnet/blob/master/data/dataset-resized.zip

https://ucsd-cse-spis-2018.github.io/lab/lab06/

https://www.tensorflow.org