RU Recycle: A Smart Self-Sorting Trash Can Using Deep Neural Network

Umama Ahmed, Steven Coulter, Jake Rodin, Yiwen Zhou
{ua66, sbc93, jar492, yz564}@scarletmail.rutgers.edu

Introduction

America accumulates more than 254 million tons of trash annually, but recycles only 34.5% of its total municipal waste. Although most are aware of the consequences of pollution and landfills, many individuals fail to recycle accurately due to lack of access or inconvenience. We aim to streamline the process of recycling with efficiency and accuracy with a self sorting trash can.

Goals

Design and build a device that will use computer vision and various sensors to categorize a disposed item and send it to the appropriate bin using a conveyor belt mechanism.

- Finetune the ResNet-50 neural network architecture with thousands of images of waste to categorize its findings into: paper, plastic, metal, glass, and trash.
- Use a Raspberry Pi 3 B+ in conjunction with an Arduino Pro Micro to control the conveyor belt system which will transport and drop the waste into the correct bin as identified by the neural network.

Research Challenges

Software Challenges:

- ❑ Due to the fact that a wide variety of items can be thrown in trash, sometimes it can be challenging even for humans to decide if an item can be recycled or not. For example, containers with food/liquid still inside should not be recycled in order to avoid cross contamination [2].
  - Solution: Add weight sensor and Volatile Organic Compound (VOC) sensor to the carrier. These sensors will help determine if food particle/excess liquid is still inside a container.

Hardware Challenges:

- ❑ Conventional trashcans available in stores are not large enough to five types of trash in a realistic situation.
  - Solution: merge two trashcans together to give more room to each compartment.
- ❑ The synchronization of the stepper motors caused a big challenge in the hardware side: in order to make them turn in sync they had to be driven from the same Arduino pin. This caused issues, as one stepper driver originally did not use as much current as the other, causing issues in the feed torque and forcing the carrier to get turned along the rods.
  - Solution: tune the current output of the steppers using a multimeter.

Future Work

- ❑ Use a more powerful processing unit than Raspberry pi to decrease operation time at the software level.
- ❑ Build a bigger acrylic carrier to allow for larger items to be disposed.

Acknowledgement

We would like to thank Professor Sheng Wei for guiding us and providing valuable insight throughout the project. We would like the Livingston Apartment Maintenance Crew for speaking to us about the importance of recycling accurately. We would like to thank our friends from various departments for their constant support. We would also like to thank The ECE department for providing us with the necessary resources to complete our project.

Hardware Methodology

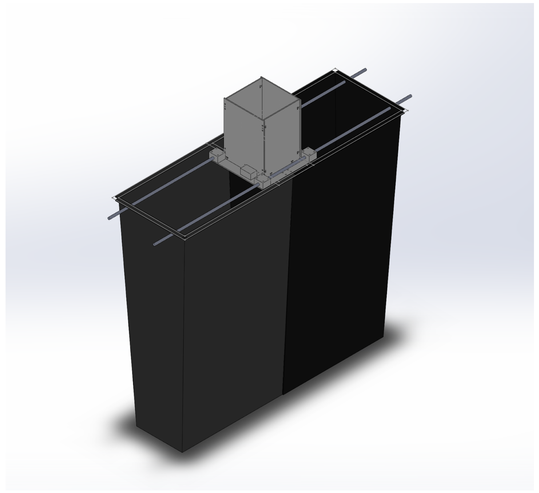

Figure 1: Solidworks design of trash can with the carrier attached on two linear rods

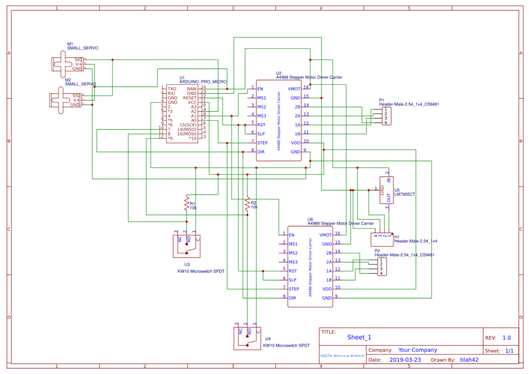

Figure 2: EasyEDA design of the PCB circuit board which communicates with the Raspberry Pi to control the stepper motor

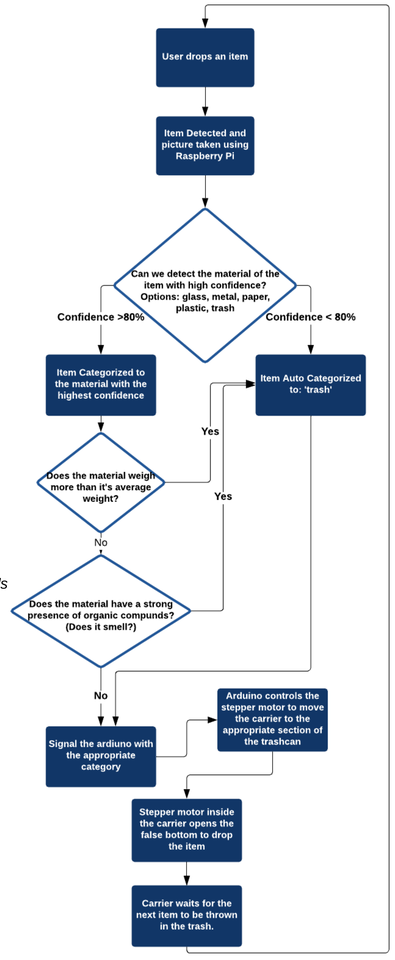

Figure 3: Flowchart of the entire process

Results

Figure 7a: Trash can full view

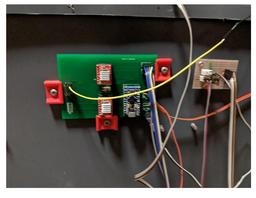

Figure 7b: Conveyor belt control circuit

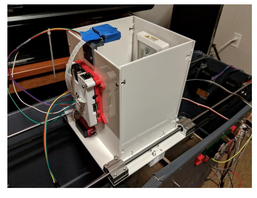

Figure 7c: Acrylic carrier

Figure 7d: Carrier top view with trash inside

Figure 7e: Carrier top view with false bottom open

Software Methodology

- ❑ ResNet-50 finetuned with TrashNet [1] consisting of 2527 images:
  - ResNet uses network layers to fit a desired underlying mapping. Because identity function is easy for a residual block to learn, it allows the user to add more and more layers to the training blocks without degrading quality or increasing training error. [4]

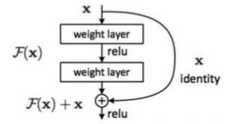

Figure 4: ResNet-50 architecture

```
#Load model
base_model = ResNet50(include_top = False,weights="imagenet",input_shape=(input_size,input_size,input_channels)
#Flatten input and add a fully connected layer
x = Flatten()(base_model.output)
x = Dense(len(classes),activation="softmax")(x)
finetuned_model = Model(base_model.input,x)
```

```
#Freeze all layers except the last 25
for layer in base_model.layers[:-26]:
    layer.trainable = False
finetuned_model.load_weights("all_nontrainable.h5")
```

Figure 5: Code snippets from the training function.

```
history = finetuned_model.fit_generator(batches, steps_per_epoch=num_train_steps, epochs=30,
 callbacks=[early_stopping, checkpointer], validation_data=val_batches, validation_steps=num_valid_steps)
finetuned_model.save('resnet50_final_epoch_30.h5')
```

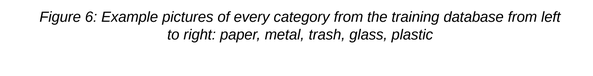

Figure 6: Example pictures of every category from the training database from left to right: paper, metal, trash, glass, plastic

Technologies Used

Materials	Precision	Recall	F-1 Score	Support
Glass	0.73	0.84	0.78	32
Metal	0.8	0.52	0.53	31
Paper	0.76	0.7	0.72	31
Plastic	0.62	0.52	0.56	31
Trash	0.42	0.58	0.5	31
Avg	0.7275	0.645	0.6475	31.25

Figure 8: Precision, recall, f1-score, and support (number of data per class) of validation data of each category [3]

References

[1] Gary Thung Mindy Yang. "Classification of trash for recyclability status". *CS229 Project Report 2016*, 2016. [Online] Available: https://github.com/garythung/trashnet. [Accessed: Apr. 18,2019].
[2] Recyclebank, "Why Can't I Recycle Stuff with Food On It", *Recyclebank*, Sept. 10,2014. [Online] Available: https://livegreen.recyclebank.com/column/because-you-asked/why-can-t-i-recycle-stuff-with-food-on-it. [Accessed: Apr. 18,2019].
[3] Jason Brownlee, "A Gentle Introduction to Learning Curves for Diagnosing Machine Learning Model Performance", Machine Learning Mastery, Feb. 27, 2019. [Online] Available: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/. [Accessed: Apr. 18,2019].
[4] Prakash jay, "Understanding and Implementing Architectures of ResNet and ResNeXt for state-of-the-art Image Classification: From Microsoft to Facebook ", Medium, Feb. 7, 2018. [Online] Available: https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624. [Accessed: Apr 19,2018]