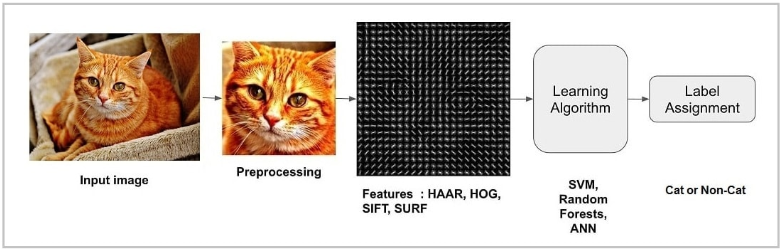
# Background

Recycling is one of the biggest problems we face today. Of the 8.3 billion metric tons of waste that is produced each year, 6.3 billion metric tons is plastic waste. Of that, only 9% is recycled. The vast majority, 79%, accumulates in landfills or sloughing off in the natural environment as litter (National Geographic). This is a tremendous amount of recyclable material, that is not being recycled. I cannot understand how people, knowingly or unknowingly, can put recycling in the garbage, and that it adds up to 79% not being recycled! If we are even able to make a tiny dent in that number, that would mean that we are causing a tremendous amount of recycling to happen. One of the multiple reasons that people don’t recycle is that they just don’t know what is recyclable! Another reason is that they are not engaged and do not realize the impact if their actions. If we are able to create a way to teach people what is a recyclable and what is not then we can improve their engagement and effectively increase the recycling rate.

# Engineering Goal

My engineering goal is to create an easy-to-use working application that can run on a mobile device that can differentiate recyclables, compost, and landfill. This way we can help guide people not knowing what is a recyclable and what is not. If we take the help of a mobile devices that are now very widely used, we can reach the majority of the people with a simple application that would be able to tell the difference between recyclables and non-recyclables. We are then accessing most of the people with such a freely deployed application. This could be used to teach and inform people about waste, so they can not only use it, but it can teach them in the long run! The end goal of such an effort is to improve community interest and engagement in recycling.

# Image Recognition and AI



# Materials

* Computer (to code the app)
* Xcode (to make, build, and test the app)
* Fruits (to test the app)
* Various recyclables (to test the app)

# Procedure

1. Learn the computer language Swift and how to develop a mobie application code with a machine learning (ML) image classifier
2. Gather pictures and datasets to train the image classifier
3. Train the classifier and create the ML Model
4. Use the ML Model and test on other items
5. Get results and train image classifier, repeat until satisfactory
6. Build a guided user interface (GUI) and a mobile device application (app)
7. Test the app by using it prospectively on a certain number of everyday objects and collect results. Evaluate accuracy and report. Modify to yield acceptable error rate, >90%.

# Citations and Source Code

1. Laura Parker, 12-20-2018, "A whopping 91% of plastic isn't recycled," National Geographic News, <https://news.nationalgeographic.com/2017/07/plastic-produced-recycling-waste-ocean-trash-debris-environment/>
2. Lowering Our, xx-xx-xxxx, "Recycling Guidelines," No Publication, https://wasteindustries.com/residential/recyclingguidelines

Source code at:

<https://github.com/Hacker719/RecycleSmartApp> and <https://github.com/Hacker719/RecycleML-Data>

# Future Improvements

The immediate improvements I would like to make to this project in the future are,

1. Build a better user interface, so it could be used and understood better across a more diverse amount of people.
2. Increase the number and diversity of the learning dataset images to make the machine learning model more accurate and make sure that it does not make people throw away recyclables or put them in the wrong bin.
3. Release the app into the app store for many people to use, and hopefully make a difference in the community! Ideally, an app could be deployed by the local recycling department for increased adoption.
4. Add a feature where it tracks which recyclables you use and your total carbon footprint, to make people more aware of what they throw away.

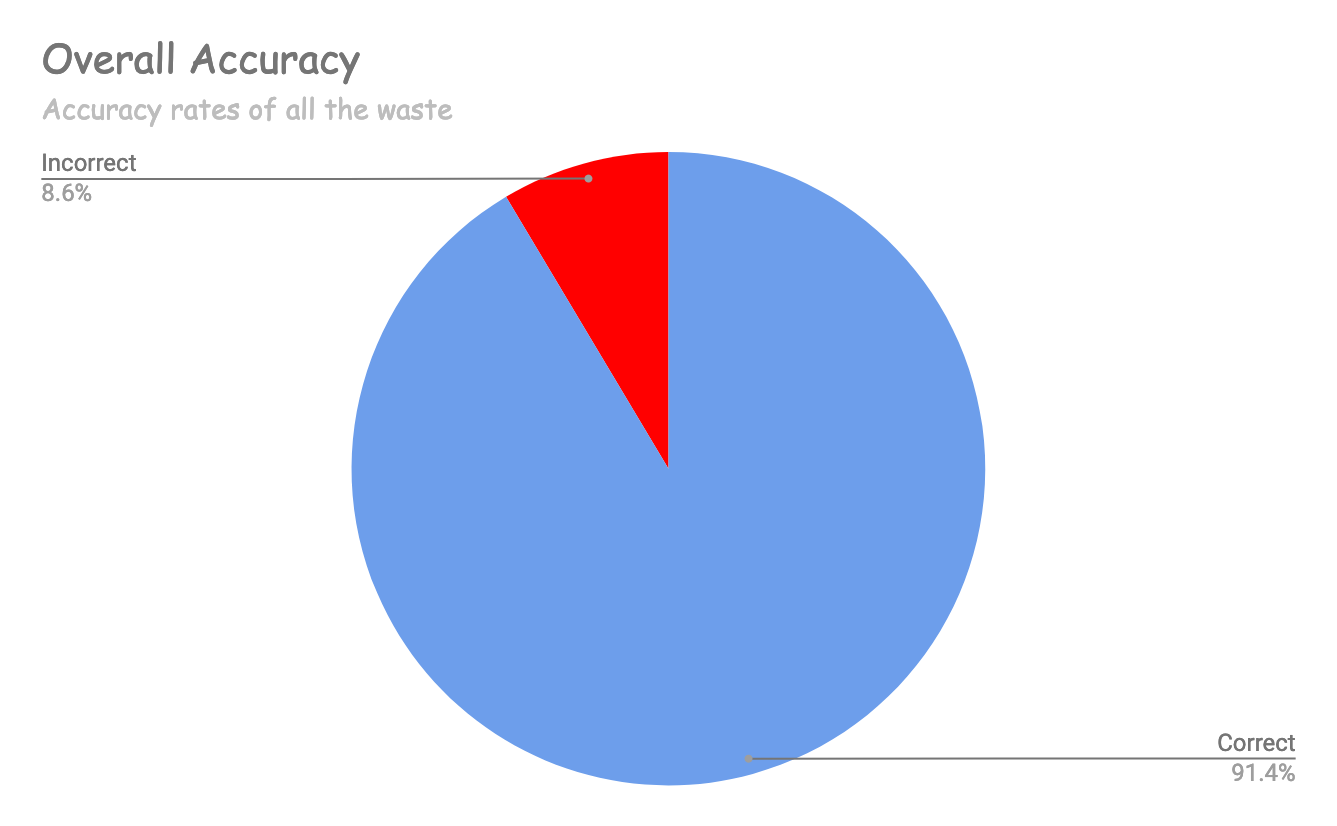
# Data

In the data analysis, we see that the app is accurate with 70 real life image tests, with an accuracy rate of 91%. Individually, the worst doing categories are the paper and glass, with 80% accuracy rates. Second comes plastic and compost, with 90% accuracy rates. The best doing ones were metal, cardboard, and trash, with 100% accuracy rates. Some ways the data could be wrong/skewed is that, it could be too small of a sample size or the lighting was different and shadows.

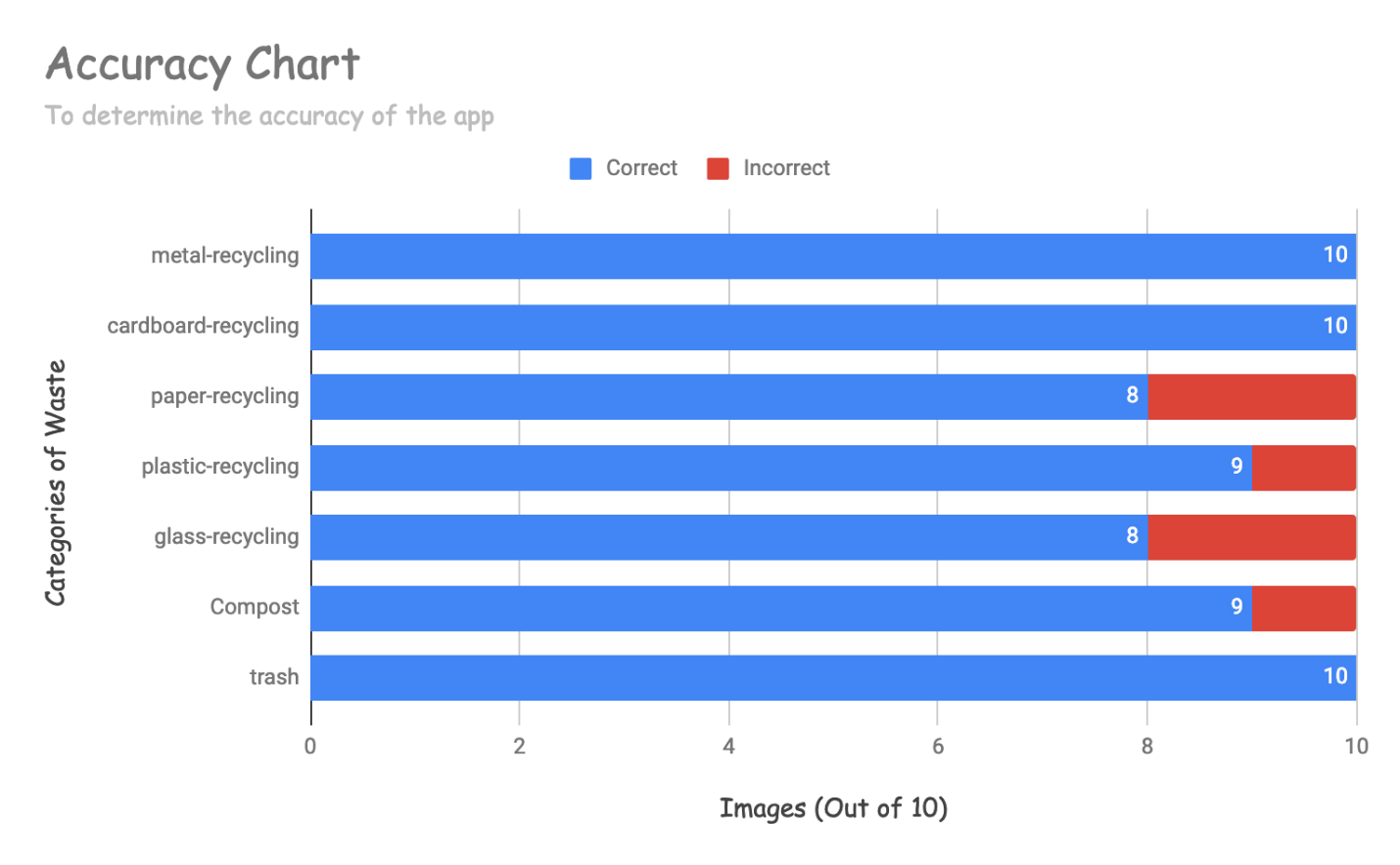
Errors can be classified as two types: Type I errors and Type II errors. Type I errors are false positives – where an item is classified as recyclable, when really it is not. Type II errors are false negatives – where a recyclable item is discarded as trash when it is indeed recyclable.

In different testing situations, one type of error is far more important to eliminate than the other. For our situation, we need to minimize Type I errors more so over Type II errors as having a nonrecyclable item in the bin is a lot more effort to identify and isolate rather than losing a recyclable item to trash.

So far there are zero Type I errors and all the other errors are Type II.



6



(It would be great if you can identify out of the error rates did you see Type I or Type II errors and what were the percentages)

(you can also show app screen shots for example on the Board – one for each of the types. Like home page, choices etc.)

# Conclusion

In conclusion, my engineering goal was completed. I was able to make a simple, usable mobile app that can be used to determine what is recyclable with a 91.4% accuracy rate. This proves that CoreML and my image dataset are a good start for deployment and can be used to make more ML models in the future.

# Further Research

I was very interested in machine learning over the course of this engineering project. I researched a little more onto how the machine learning algorithm was made. Machine Learning algorithms usually graph data from images and create boundaries between them. For example, it could graph the intensity of each pixel or the color. For 2 inputs (say color and light), it is easy enough and on a two-dimensional (2-d) plane. But with multiple independent variables, for example if there are 3 variables, you would need a 3-d plane, or for 4, a 4-d array and so on. This quickly gets hard for the human mind to graph and draw boundaries between to be accurate. This is the reason why we input the data into a computer and the computer graphs it and using data, calculates a boundary with the least error rate.

