Automated Recycling System Using Computer Vision

Desi Tomaselli

ECE-499: Capstone Design Project

Advisor: Prof. Cotter

March 26, 2020

1 Summary

There are many underlying issues with pollution and the disposal of waste. So the goal of this project was to design and build a system that would serve as a tool to help humans recycle properly in order to reduce pollution and waste. This system must be able to classify waste objects into 4 separate categories: glass, metal, plastic and paper with an 80% success rate and within 1 second. In order to achieve this, a pre-trained convolutional neural network was used on Google Colab, which is a cloud-based service provided by Google. Resnet34 was the pre-trained convolutional neural network that was used in my project. There are 34 layers to it with a majority of them being convolutional layers. The two datasets that were used is TrashNet (provided by Gary Thung on Github) and my own personal one that I created. The TrashNet dataset consisted of 2527 images in total with 6 different categories: metal, glass, plastic, paper, cardboard and trash. However, the dataset was modified so that it only included the images among the 4 categories that were stated above, which consisted of 1987 images in total. I was able to achieve an 86.7% success rate using the TrashNet dataset and a 68.0% success rate on my personal one. The model did really well identifying glass, paper and metal using the TrashNet dataset, but it had some trouble trying to identify plastic due to the variation of plastic that had very similar features to glass, paper and metal. It did well on identifying paper and plastic using my dataset, but had trouble with glass and metal because of the brightness levels in some images and because some glass images had features that looked very similar to plastic. Hopefully, by including more variation of images into my personal dataset, it will improve the success rate and then a physical trashcan can be assembled and the entire system could be implemented and used in the real world.

Contents

List of Figures

2 Introduction

Garbage is a global issue that all living species are affected by. It has become a considerable issue the past decade due to garbage and pollution [1]. More specifically, humans have been dumping plastics and other waste into oceans to create these massive garbage patches in them. They are becoming so big that the “Great Pacific Garbage Patch (GPGP) is a massive area of floating waste that is more than twice the size of Texas” and “it contains about 1.8 trillion pieces of plastic” [2]. This dangerous number is between “4 and 16 times the mass of plastic that scientists previously estimated” [2]. To make the issue even worse, it is growing in size, exponentially [2].

However, this is not the only garbage patch that exists. This is “one of five ocean garbage patches” that’s grown in size around the world “as people use more and more plastic, which is not biodegradable and is used for everything from water bottles to shipping crates” [2]. This has become such a prominent issue that even researchers who were not initially going to conduct a study on plastic pollution, ended up doing so because they were so alarmed by the amount of plastic that got entangled on their devices for research on plankton. [3]. Research was initially conducted in 1959 to measure plankton in the north Atlantic Ocean. Since then, devices and gear to measure the plankton has been tangled up with plastic almost “669 times” and since then the occurrence has “skyrocketed in the past two decades” [3]. Scientists specifically say the occurrence “has increased by around ten times” [3].

With these staggering numbers and the ocean patches growing exponentially in size, it is clear that action needs to occur to fix the issue. In fact, research shows that “plastics in the ocean kill or harm more than 300,000 marine animals every year” [4]. And because of this, it will become the “deadliest threat to [humanity]” if people do not rally together to fix it since millions of people depend on the oceans [4]. Therefore, humanity needs to start fixing the issue and the first step to that is by preventing it from getting worse. One solution to this is by having people recycle properly. Proper recycling takes time and effort from the individual, but it is important for the environment as it helps reduce pollution caused by waste and the need for raw materials [5]. But the issue with recycling is that it is a manual process and not many people think about it and when they do, they do not do it correctly [6].

Therefore, this paper proposes a solution of an automated recycling system (ARS) using computer vision to fix this global issue. The computer vision technique that is used is convolutional neural networks (CNN), which will be discussed later in the paper (6.1). The goal is for the system to classify waste objects into 4 separate categories: glass, plastic, metal and paper with an 80% success rate and within 1 second (4). In order to achieve this, it requires a CNN model, Resnet34, (6.2) to train and learn on datasets of images such as TrashNet (6.3) and my own. These datasets and the model produced very interesting results which can be found here (7).

3 Background

3.1 Classifications for Recycling

There are multiple classifications for recycling. They usually consist of metal, glass, plastic, paper, cardboard, and organic waste. However, depending on the country, these classifications may vary. For example, in Japan, they have one of the most complicated recycling systems in the world. They have bins dedicated to plastic bottles, plastics, recycled papers, cans, glass bottles, burnable garbage and non-burnable garbage [7]. Whereas in the United States, the most common ones are plastic bottles, paper and cardboard, metal cans, glass bottles, compost and other waste [6]. However, because of this, it makes recycling inconsistent. Therefore to create a practical system, it is important to determine classification standards of recycling so that anyone around the world can use it. This topic will be discussed more in the design requirements section (4.1.1).

3.2 Sensors

In the past, many people have conducted experiments and studies on the topic of object recognition systems to create waste segregation systems. The methodologies that were used vary greatly since there is not one solution. These methods involve different kinds of sensors to determine objects' conductivity [8], weight and density characteristics [9]. Some people even used light sensors to measure the color of objects by sending photons at them and sensing the light spectrum of colors that were being reflected [9], while others used cameras as sensors to determine the features and contours of objects and shapes in an image [10], [11], [12]. In the design alternatives section (5) and the design section (6), I go into more detail about what specifically people used and how they were able to use them.

3.3 Computer Vision

For the people that used cameras as their sensors, they used computer vision to help them identify objects within images. Computer vision is a computer science concept that enables a computer to identify objects autonomously. Many companies like Tesla are utilizing this technique [13] to create self-driving cars since the car will need to have the ability to identify people and objects while on the road. They used and considered many concepts that I will be mentioning in my paper such as grayscale, threshold, SIFT, and KNN to name a few that helps identify people and objects. These methodologies will be discussed in more detail in the design section (6) since this is the route that I decided to take.

3.3.1 Image Processing Methods

A subsection of computer vision involves some image processing techniques in order to alter the image. Image processing is very important for extracting features within an image as it makes it easier for computers to differentiate objects from noise. Some basic image processing methods include grayscale and thresholding.

3.3.2 Feature Extraction

In computer vision, feature extraction is used to determine features within an image and to later be used to compare to similar features in other images. These methods are necessary in the identification process of objects since the comparison of features will be a major reason as to why one object in an image is classified as its classification. Some feature extraction methods include SIFT and Hu moments that many people have previously explored [14], [1], [10], [12]. These methodologies will be properly discussed in the design alternatives section (5) and the design section (6).

3.3.3 Classification Methods

What comes after feature extraction is classification. There are multiple methods in classifying objects. These classification methods are what compare features together to classify objects. The methods that will be discussed will be the k-nearest neighbobr (KNN) algorithm, the support vector machine (SVM) algorithm and convolutional neural networks (CNN). People in the past have used these methods in their own systems to classify objects [15], [16], [10], [11], [12], [17] and they were able to get some promising results. These methods will be discussed in more detail in the design (6) and design alternatives (5) sections below.

3.4 Processing and Computing Devices

All of the methodologies (sensors and computer vision algorithms) that were discussed above need a computational device that is capable of meeting their computational needs. For instance, capacitive or light sensors [8] or even machine learning algorithms [1] in conjunction with a microcontroller may be the only computational device necessary for the system whereas a convolutional neural network (CNN) requires more computational power [14], [18] involving a computer with a high clock speed for its central processing unit (CPU) and a dedicated graphics processing unit (GPU) with several gigabytes of RAM. CNNs are more computationally complex compared to machine learning algorithms. Therefore they tend to achieve higher accuracy rates [14], [10].

3.5 Datasets

Machine learning algorithms and CNNs both require datasets to train on. Therefore choosing the right dataset for a specified project is very important as it will influence the way these methods identify and classify objects or people. Some datasets were posted for public use such as TrashNet [19], [12], [18] while others were privatized [16]. These privatized datasets were used only for their associated experiment. But all datasets can be time consuming to create as it usually requires 100s of images at a time per classification [12]. For CNNs it usually requires many more images since they really benefit from training on bigger datasets.

4 Design Requirements

4.1 Classifier System

4.1.1 Classifications

The classification system should be able to identify and classify objects that are metal, plastic, glass and paper. These four categories were chosen for simplicity reasons [10]. In other papers such as Yang's [12] and Aral's [18], they used six different classifications, glass, paper, metal, plastic, cardboard and trash since they both used the same dataset, TrashNet [19]. However, it is important to know that they did not tackle the problem of trying to separate their classified objects into six different containers. However, the ones that did [11], [1], used minimal classifications. In Garcia's paper [11], they only had three types of inorganic waste to classify which was plastic bottles, plastic cutleries and alumnium cans. In Salmador's paper [1], they classified objects that were metal, glass, plastic and paper. Since this was the case, I decided to choose the middle ground between the previous projects that were created and determined to only classify objects that were metal, plastic, glass and paper since those were the most common and it was not too many classifications to try to separate into.

4.1.2 Speed

The system should be able to classify objects into the 4 separate categories within 1 second. If it is within 1 second, then it makes the system practical to use. It is not practical if the classification process takes several minutes because if the system were implemented to be used in real life, then the system would need to be fast enough to be able to classify waste in time for more waste to enter the system. Therefore, it needs to be within this range of speed in order stay efficient and practical.

4.1.3 Accuracy

The success rate of the system should be at least 80% or higher. In Silva's paper [10], they were able to achieve 93% with one of their pre-trained convolutional neural networks (VGG16). Also if the success rate is below 80%, then it will not be practical to use since it would be much easier to recycle manually rather than rely on a system to do it for us.

4.1.4 Power Consumption

The power consumption of the device should be relatively low. One of the low power consuming devices that exist is the Raspberry Pi 3 Model B. According to Raspberry Pi Dramble [20], A Raspberry Pi 3 Model B should consume 260mA (1.4W) or less while idle and under 400% CPU load, it should consume 730mA (3.7W). If a battery were to be attached to this device, it would not be practical for it to contain only one AA battery [21] as the machine would probably die after 9.2 hours while idle and 3.3 hours while under heavy load [22]. It would also not be able to sustain for very long even with a laptop battery since they tend to be 3x bigger [23]. Therefore, it is unlikely that the attachment of a battery will occur, even though it would make the system more practical since it would allow for portability. This makes it more useful in the real world since it would not need to be continuously connected to a wall outlet in order for it to function.

4.1.5 Cost

The overall system should relatively cheap. The cost should be less than $100 so that it's accessible to public facilities such as schools, libraries, hospitals, police stations, malls, etc.... With a more expensive system, it'd be difficult to manufacture and mass produce since it could be very costly if they were to be bought in bulks.

4.2 Segregation System

Although I was not able to implement the segregation system, some thought was put into creating one. If one was created these are the design requirements that it would need to have.

4.2.1 Power Consumption

The power consumption of the device should be relatively low so that it does not consume excessive raw resources. A couple of motors should not draw an exceeding amount of power in relation to the Raspberry Pi. However, like it was mentioned above, the likelihood of attaching a battery to the system is unlikely since the system would only last a couple of hours at most.

4.2.2 Speed

The speed of the segregation system should be as fast as the classification system is. If the segregation system is slower than the classification system then it will be bottle-necking the entire system, which is not desirable.

4.2.3 Accuracy

The accuracy of the segregation system should correspond to whatever the classification system's accuracy is since that system is the responsible for identifying objects. Therefore, there should not be any inconsistency between the two systems. In other words, the segregation system should have an accuracy rate of 100% in relation to whatever declaration is made about object by the classification system.

5 Design Alternatives

5.1 Hardware

5.1.1 Sensors

When the project was initially being developed, the overarching question to be answered was how to classify objects. There are two possible solutions to this problem. One was to use sensing to identify and classify objects. These sensors could be capacitive, inductive, infrared, proximity, light [9] and ultrasonic sensors. Depending on the sensor, it could be determined what the object was made of based on the value that it returns [8]. For example, an inductive sensor is great for detecting metals because it generates an oscillating electromagnetic field and can detect the change in the field due to the metallic object. Or a conductive sensor (Figure 1) measures the electric current between two electrodes, and it sends a current through a given material. The more conductive the material is, the higher conductivity values it produces [8]. Therefore, the values within certain ranges could be used to determine whether an object is metallic or not.

Figure 1: Capacitive Sensor

However, despite knowing these facts about sensors, after doing intensive research, I realized that recently more people used computer vision rather than sensors to achieve their classification and identification of objects [10], [16], [18], [14], [11], [15]. They were also able to achieve better results at the lowest cost across the different classes using computer vision compared to sensors so I decided to use computer vision for my methodology on this project.

5.2 Software

As it was mentioned in the background section 3.3, people also use cameras as their sensor and they rely on computer vision techniques to develop object recognition systems to identify and classify objects. Below I will be mentioning the alternative computer vision techniques that I explored, but did not end up pursuing for my system.

5.2.1 Feature Extraction

Hu Moments: One feature extraction method that was mentioned a few times in papers was Hu Moments [1], [11]. Generally, the way Hu Moments works is that it requires a grayscaled image and it uses the concept of thresholding to separate the object from its background [24]. It does this by taking the gray values of pixels within an image and it determines the "moments" of an object by taking the integral of it [24]. Several ordered moments can be calculated by taking multiple integrals of the object, which leads to spatial, central, and central normalized moments. These moments can be compared to other moments of other objects in images and that is how identification occurs using this method [24].

I decided to not use Hu Moments because SIFT is the most efficient local descriptor and it performs the best [25], [26]. "Moments... show the best performance among the low dimensional descriptors," [25] but "SIFT gives the best recognition rate" [26]. Therefore, I wanted to use SIFT over Hu Moments.

SIFT: SIFT stands for Scale-Invariant Feature Transform which is an algorithm that is commonly used for detecting features within images. It is often used because like the title suggests, it works for objects in images of multiple scales, making it practical to use. The way SIFT works is that it takes keypoints from another algorithm and finds patches of pixels around those given keypoints [26]. These patches consist of pixel gradients within the image [27]. These gradients are then used to create orientation historgrams - also known as keypoint descriptors [26]. These orientation histograms contain important aspects of the patches of pixels. These descriptors achieve robustness against contrast changes, and rotations [28]. Figure 2 shows a visual representation of SIFT and a more detailed explanation of what SIFT is can be found on openCV's documentation website [28].

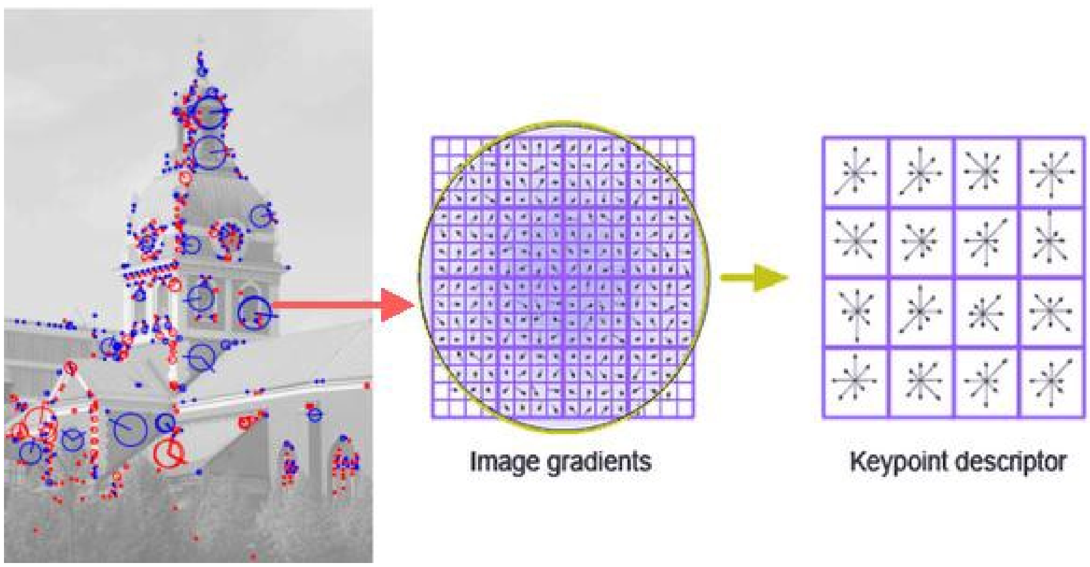

Figure 2: Concept of SIFT [27]

5.2.2 Classification Methods

KNN Algorithm: K-Nearest Neighbors is an algorithm used for image classification [29]. Due to simplicity, performance and computation requirements, this algorithm was very desirable. People in the past were able to get results above 80% [10]. The purpose of the algorithm is to train it on a dataset of images and get it to recognize and classify objects depending on their features and characteristics that were defined for them by feature extraction methods. Then the new object is classified to its most common class defined by the majority of its k nearest neighbors to it by calculating the Euclidean distance between all of its points [10]. This concept can be seen in Figure 3. Then the algorithm returns the percentages of every category that the object in the image could be.

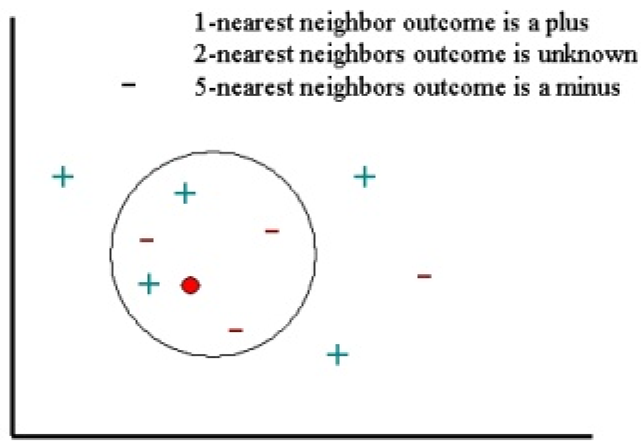

Figure 3: Concept of KNN [30]

The figure above shows KNN works conceptually. If k=1, then the red dot would be classified as a plus due to a plus object being the closest to the red dot. If k=2, then the red dot would be unknown because the two closest objects to it are a plus and minus. And the figure above displays the scenario of k=5. If k=5, then the red dot would be classified as a minus object because although it is considering the 5 closest objects in hyper-plane, among those 5 objects are 3 minus objects. Therefore, it'll classify the red dot as a minus.

This algorithm is perfect for my system because it is simple and easy to implement [11]. It also has good performance (as mentioned above) and it has low enough computational requirements to apply it to a microcontroller such as a Raspberry Pi 3 Model B. However, due to time constraints, I was not able to fully implement the algorithm.

5.2.3 Classification Methods

Support Vector Machine: One popular classification method that has been mentioned a few times in papers [12], [1], [10] is the Support Vector Machine (SVM) algorithm. SVM is a machine learning algorithm that's used to identify and classify objects. It does this by separating objects on hyperplanes for multi-dimensional data. Figure 4 depicts this concept. A more detailed explaination can be found here [31], [26] and here [12].

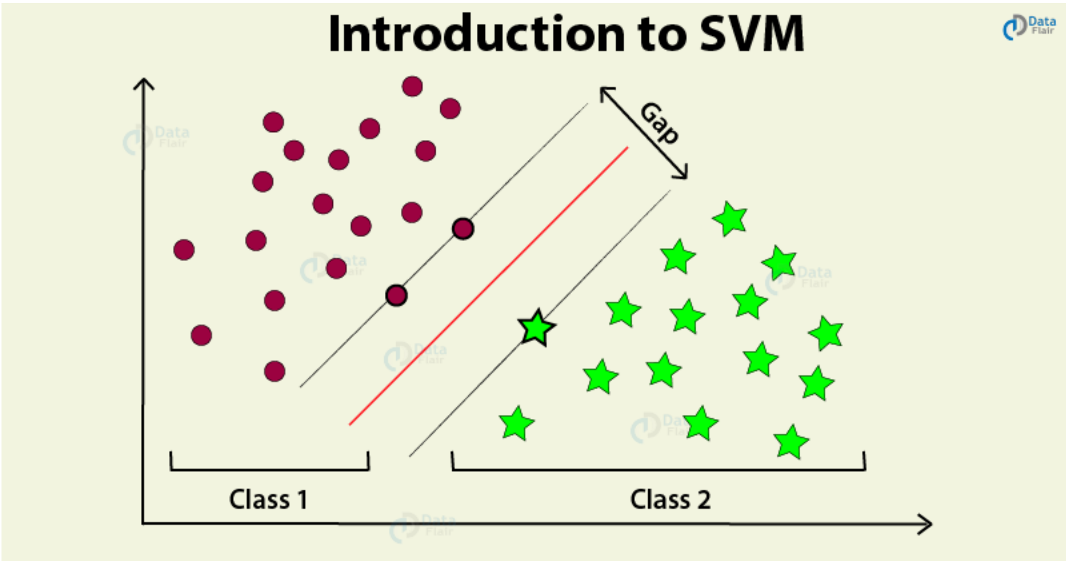

Figure 4: Support Vector Machine (SVM) [31]

It was decided that it would be best to not use the SVM method for this project. Due to Salmodor's research project [1], he explains that classes can be "quite heterogenous" meaning that they can overlap and a "partitioning algorithm like SVM is not a good option." SVM also scored a lower accuracy rate than KNN according to Silva [10]. Therefore, this lead me to conclude that SVM is not the classification method that I want to use for this type of project.

6 Design

The overall design of the classifier system consists of using convolutional neural networks (CNNs) to classify objects in images. The convolutional neural network that I used for my project is known as Resnet34 (6.2). The dataset that I decided to use is TrashNet (6.3).

6.1 Convolutional Neural Networks

A convolutional neural network (CNN) is a very popular method used for image analysis and image classification [10], [17]. These neural networks model the human brain (its neurons and receptors) to replicate the process of learning in a human being. Figure 5 shows a depiction of what a neural network looks like.

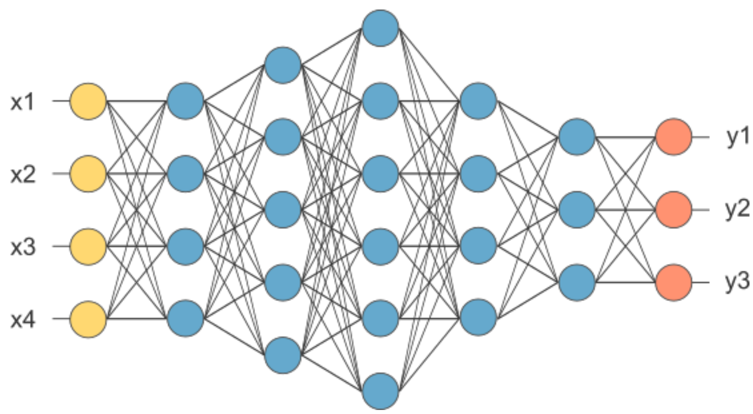

Figure 5: Fully Connected Neural Network [32]

These networks are a large segment of computer vision and are an effective and efficient way to give machines identification capabilities. CNNs differ from regular neural networks because they have convolutional layers in the network [32]. These convolutional layers are able to detect patterns within an image. With each convolutional layer, there are a certain number of filters or kernels that need to be specified and defined. While the first layer involves basic filters to extract features and shapes, the deeper layers may be more specific and exist to detect specific objects or regions in an image [32].

Filters, also known as kernels, are made of n by m matrices and they are initialized with random numbers. When a filter is given an input, the filter slides across the input over an n by m set of pixels until it has slid over every single set of pixels in an image. This is known as convolving.

Convolution is the concept of multiplying individual pixels of an image with their corresponding pixels in a filter and adding all the products with each other until the entire set of pixels have been summed together. Then it places the resulting pixel in its corresponding location within the convolved image.

For example, seen in Figure 6, a 3 by 3 filter is being applied to the source image where the top pixel, (3) is being multiplied with the top pixel of the filter (-1) and then the result of that is summed with the product of the pixels next to it which are 0 and 0 (from the source image and the filter). And it continues until the entire set of pixels have been added together and the resulting pixel goes to the destination pixel of the convolved image.

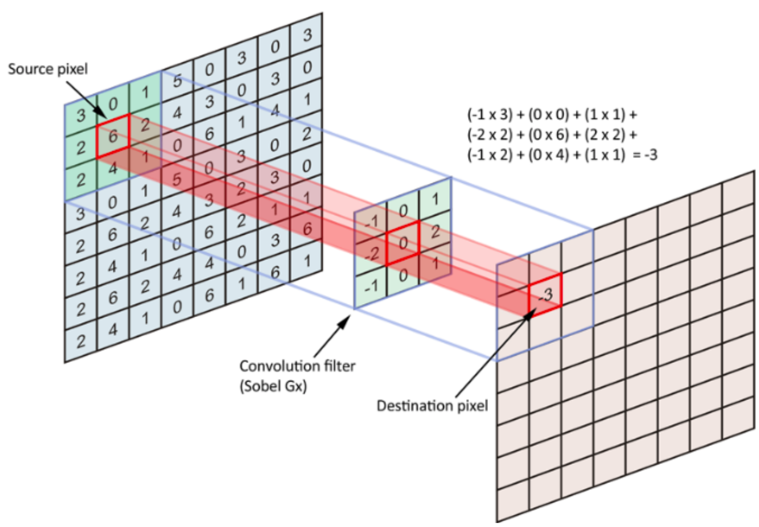

Figure 6: Convolving a Filter over an Image

The result of convolution allows for the network to able to detect patterns and features within images. After the image goes through every layer in the network, the object should be identified and classified.

6.2 Resnet34

Resnet34 is a pre-trained convolutional neural network. This means that it has already learned some visual features from previous trainings. Pre-trained models typically outperform non-pre-trained models [33] which is why I decided to use it over others. Resnet34 has 34 layers in it and many of them are convolutional layers. The model can be seen in Figure 7.

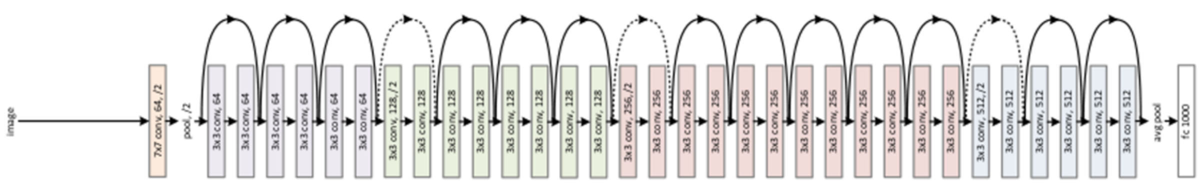

Figure 7: Convolutional Neural Network: Resnet34

6.3 TrashNet Dataset

The dataset that was chosen for this project is called TrashNet, which was created and provided by Gary Thung on Github [19]. He made this dataset for his own research purposes. Thung and a research partner of his, Mindy Yang, [12] conducted an experiment to explore the possibility of classifying trash into different recycling categories through various machine learning and computer vision algorithms. This dataset provides 2527 images in total with 6 different categories: metal, glass, plastic, paper, cardboard and trash. Every category has roughly 450 images each. However, for this project, I modified the dataset to use only 1987 images out of the 2527 images in order to classify only 4 categories: metal, glass, plastic and paper. Below are sample images from the dataset - one from every category.

   

I decided to split the dataset into a training dataset (75%), validation dataset (12.5%) and test dataset (12.5%) for the CNN. The training dataset is used to train the model and get the model to recognize patterns and features within the images. The validation dataset is used to determine if the training should be stopped. Finally, the test dataset is used for the model's predictions to see how well the model correctly classified images based on the features and patterns it detected from the training dataset. Below, figure 8 depicts the dataset being fed into the model.

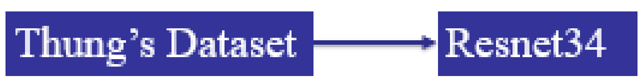

Figure 8: Thung's dataset being fed into the model

6.4 Google Colab & Google Drive

The platforms that I used to host all of these components for the system are Google Drive and Google Colab, which are cloud-based services that are provided by Google. These fast services allowed me to store and access the entire TrashNet dataset and access Resnet34 from any computer very easily.

Google Drive provides the user with 15GB of free storage and Google Colab allows the user to use 12GB of RAM and gives free access to a remote Telsa K80 GPU if the user needs more computational power. The Tesla K80 has 24GB of RAM, 480GB/s of memory bandwidth, and 4992 CUDA cores. This GPU can be used for computer vision techniques such as CNNs since it can easily handle the high demand that a CNN would computationally require. It allowed me to train the model much faster. For example, with the GPU, the runtime per batch of images that the model trained on would take 45 seconds each. But without the GPU, the runtime per batch of images would take up to 6 minutes. And because of this, the process of training and validating the model was quick and easy.

7 Results

7.1 Results Thung's Dataset

The success rate that I got with Thung's dataset was 86.7%. This accuracy rate is higher than 80%, which was a performance goal. I decided to take a look at the confusion matrix to see what the model had trouble with. A confusion matrix is used to describe the performance of a model on test datasets where true values are known. So the confusion matrix for these results are shown below in Figure 9.

	Glass	58	1	0	4
	Metal	1	48	0	2
Actual	Paper	0	0	73	2
	Plastic	2	7	14	37
		Glass	Metal	Paper	Plastic
			Predicted		

Figure 9: Thung's Confusion Matrix

Based on this matrix, it can be seen that the model did really well identifying glass, metal and paper. However, it had some trouble trying to identify plastic. The model thought 14 images of plastic were paper, 7 images of plastic were metal and 2 images of plastic were glass. When looking back at the dataset, it makes sense as to why this occurred based on a few of these plastic images.

The image all the way on the left looks like glass. The image right next to it looks like paper. And the images to the right look like metal so it is understandable as to why the model thought these images might be classifications other than plastic since they resemble similar colors and shapes to them.

(a) (b) (c) (d)

Figure 10: TrashNet's Plastic Images: (a) Plastic like glass (b) Plastic like paper (c) Plastic like metal (d) Plastic like metal

7.2 Results with 'Real' Test Dataset

The next steps after meeting the success rate goal were to create a test dataset of waste myself. Therefore, the model was tested on this dataset, which mimicked a real world scenario where a collection of new images that came from somewhere else would be fed into the model. So approximately 68 photos of waste were taken (17 for each category) and they were loaded into the model. The success rate was 68.0%. In order to figure out why the success rate was much lower than the rate from before, the confusion matrix was looked at (Figure 11).

			Real Test Dataset		
	Metal	11	6	2	0
	Paper	0	14	0	0
Actual	Plastic	0	1	19	0
	Glass	4	2	7	2
		Metal	Paper	Plastic	Glass
			Predicted		

Figure 11: 'Real' Test Dataset

Based on the matrix, it seems as though the model was able to properly identify paper and plastic. However, it had some trouble trying to identify metal and it drastically struggled with identifying glass. For some reason the model thought that 7 images of glass were plastic, 2 images of glass were paper and 4 images of glass were metal, which is a little strange because the model was able to identify glass with no problem using the previous dataset. So the images of glass in both datasets were compared to each other.

7.3 Analyzing The Results

With Thung's modified dataset, a 86.7% success rate was achieved compared to my own test dataset, which had a 68.0% success rate. When looking at the confusion matrices, it seems as though the model was correctly identifying glass images for Thung's dataset, whereas it struggled with mine.

Glass	4	2	7	2

(a)

Glass	58	1	0	4

(b)

Figure 12: Test Datasets: (a) 'Real' Test Dataset (b) Thung's Test Dataset

So after some investigation, I found some major differences between the two datasets with their glass categories. One major difference that I noticed is that in Thung's dataset, there are many images of wine glasses and there are practically none in mine (Seen in figure 13).

Figure 13: Wine Glass

Also I was curious to see how well the model would do with images of objects in low light settings so I changed the lighting environment when I took my pictures (Figure 14).

Because of these darker images the model did not do well with them. There were a total of 22 images that were darker (not well illuminated) than the rest. 8 of the glass images and 3 of the metal images were dark and they were not correctly identified. However, the model was able to correctly identify all the paper images despite 5 of them being dark. 6 images in the plastic category were dark, but nearly all of them except for 1 were correctly identified by the model. The entire training and validation datasets involved images that only had good lighting environments; so the model was not well prepared for darker images.

Figure 14: A wine bottle in a low lit environment

The last difference that I noticed with the glass images between the two datasets is that mine had glass images that looked like plastic due to their color and shape (seen in figure 15). This glass bottle happens to be from Japan and none of the bottles from Thung's dataset were from that country. Therefore, it is obvious that due to lack of variation of waste images in the training dataset, the model had trouble trying to correctly identify what category some waste belongs in. Not only that, but there was very little variation of glass bottles in my own test dataset. Majority of the test dataset only involved only 2 different kinds of glass bottles (the Japanese bottle and a wine bottle). Therefore, the model was having trouble with identifying the same bottles over and over, but just at different angles and with different lighting environments.

Figure 15: Japanese bottle

The overall conclusion that I can draw from these results are that in order to improve the system, more variation of waste should be included in all the datasets (training, validation and test). This includes variation in lighting-environments and the waste itself. And due to time constraints, I was unable to test for time efficiency for the model since I had to manually input the actual classifications for my individual test images.

8 Conclusions

In conclusion, the reason why this project was created in the first place was due to the concerns that have been rising in the past decade about pollution and waste. Pollution and waste have been increasing due to the lack of understanding of how to dispose of waste properly. Therefore, I proposed a solution by creating an automated recycling system in order to help us dispose of waste better. I determined that convolutional neural networks would be the best option for this project since they are the state-of-the-art technology for image classification. The results that I got was an 86.7% using Thung's dataset and a 68.0% for my own test dataset. Although the goal was to achieve at least an 80% or higher for the success rate, I am satisfied with the result that I got with my own test dataset based on knowing the difference that I found between the two datasets. My dataset used wastes from another country compared to the dataset that the model trained on and the lighting-environments in images were very different.

Something to take away from this project and the results of my project is that the number of photos and variation of photos being fed into the network is a very important aspect of convolutional neural networks. The more photos and variation in them, the higher the success rates will be. I also learned that image processing and image classification is not a trivial problem at all. In order to achieve something like this and implement it into the real world, it requires a lot of resources and thought-processes to help make it successful.

8.1 Future Work

There is a lot of future work that could be applied to this sort of application such as adding more photos to the overall dataset that the model will be handling. A physical trashcan should also be assembled and implemented along with the CNN so that it can be tested with real waste going into the system. Lastly, it would be nice to include an ”other” category since waste will most likely not fit within 1 of the 4 categories that were specified in this project. For example, compost waste would not fit into any of the 4 categories that I defined here.

References

[1] Alvaro Salmador, Javier Pérez Cid, and Ignacio Rodríguez Novelle. Intelligent garbage classifier. *IJIMAI*, 1(1):31–36, 2008.

[2] Andrew Freedman. The great pacific garbage patch is up to 16 times more massive than thought. *Mashable*, Mar 2018.

[3] Doyle Rice. Sea sick: Plastic garbage in the north atlantic ocean skyrocketing. *USA Today*, Apr 2019.

[4] Plastics pose biggest threat to oceans — africa renewal. ://www.un.org/africarenewal/magazine/may-july-2017/plastics-pose-biggest-threat-oceans.

[5] Recycling basics. `https://www.epa.gov/recycle/recycling-basics`, Nov 2019. EPA.

[6] Edward" "Humes. "the us recycling system is garbage", Jun 2019.

[7] Plastic fantastic: How does tokyo recycle its waste? `https://www.japantimes.co.jp/life/2017/06/10/environment/plastic-fantastic-tokyo-recycle-waste/#.XnKFBqhKhPZ`.

[8] Amrutha Chandramohan, Joyal Mendonca, Nikhil Ravi Shankar, Nikhil U Baheti, Nitin Kumar Krishnan, and MS Suma. Automated waste segregator. In *2014 Texas Instruments India Educators' Conference (TIIEC)*, pages 1–6. IEEE, 2014.

[9] Alexander M McQueen, Craig D Cherry, Joseph F Rando, Matt D Schler, David L La-

timer, Steven A McMahon, Randy J Turkal, and Brad R Reddersen. Object recognition system and method, May 30 2000. US Patent 6,069,696.

[10] Alexandre Silva and Eduardo Soares. Artificial intelligence in automated sorting in trash recycling. 10 2018.

[11] Andres Torres-García, Oscar Rodea-Aragón, Omar Longoria-Gandara, Francisco Sánchez-García, and Luis Enrique González-Jiménez. Intelligent waste separator. *Computación y Sistemas*, 19(3):487–500, 2015.

[12] Mindy Yang and Gary Thung. Classification of trash for recyclability status. *CS229 Project Report*, 2016, 2016.

[13] Lora Kolodny. Tesla is buying computer vision start-up deepscale in a quest to create truly driverless cars. `https://www.cnbc.com/2019/10/01/tesla-acquiring-deepscale-computer-vision-start-up-for-self-driving.html`, Oct 2019.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[15] Cheng-Jin Du and Da-Wen Sun. *Object Classification Methods*, pages 81–107. 12 2008.

[16] Param Popat, Prasham Sheth, and Swati Jain. Animal/object identification using deep learning on raspberry pi. In *Information and Communication Technology for Intelligent Systems*, pages 319–327. Springer, 2019.

[17] Yinghao Chu, Chen Huang, Xiaodan Xie, Bohai Tan, Shyam Kamal, and Xiaogang

Xiong. Multilayer hybrid deep-learning method for waste classification and recycling. *Computational Intelligence and Neuroscience*, 2018, 2018.

[18] Rahmi Arda Aral, Şeref Recep Keskin, Mahmut Kaya, and Murat Hacıömeroğlu. Classification of trashnet dataset based on deep learning models. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 2058–2062. IEEE, 2018.

[19] Garythung. garythung/trashnet, Apr 2017.

[20] Power consumption benchmarks. `https://www.pidramble.com/wiki/benchmarks/power-consumption`. Accessed: 2019-11-19.

[21] Battery capacity. `http://www.techlib.com/reference/batteries.html`.

[22] Battery life calculator. `https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-battery-life`.

[23] Asher Moses. All about laptop batteries: Updated. `https://www.cnet.com/how-to/all-about-laptop-batteries-updated/#:~:targetText=Thetypicalnotebookbatteryhas,hours(mAh)ofcapacity.`, Aug 2008.

[24] Johannes Kilian. Simple image analysis by moments version 0.2, 2001.

[25] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. 2005.

[26] Anant Choksuriwong, Bruno Emile, Helene Laurent, and Christophe Rosenberger. Comparative study of global invariant descriptors for object recognition. *Journal of Electronic imaging*, 17(2):023015, 2008.

[27] Andrey Nikishaev. Feature extraction and similar image search with opencv for newbies, Sep 2019.

[28] Introduction to sift (scale-invariant feature transform). https://docs.opencv.org/3.4/da/df5/tutorial_py_sift_intro.html.

[29] Mohd Helmy Abd Wahab, Aeslina Abdul Kadir, Mohd Razali Tomari, and Mohamad Hairol Jabbar. Smart recycle bin: a conceptual approach of smart waste management with integrated web based system. In *2014 International Conference on IT Convergence and Security (ICITCS)*, pages 1–4. IEEE, 2014.

[30] Www.statsoft.com. http://www.statsoft.com/Textbook/k-Nearest-Neighbors.

[31] DataFlair Team. Support vector machines tutorial - learn to implement svm in python, Aug 2019.

[32] Prabhu. Understanding of convolutional neural network (cnn) - deep learning. https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148, Nov 2019.

[33] Florin-Daniel Cioloboc. Why use a pre-trained model rather than creating your own? https://medium.com/udacity-pytorch-challengers/why-use-a-pre-trained-model-rather-than-creating-your-own-d0e3a17e202f, Jan 2019.

[34] Fernández Hernández, B Yumilka, Rafael Bello, Yaima Filiberto, Mabel Frías, Lenniet Coello Blanco, and Yaile Caballero. An approach for prototype generation based on

similarity relations for problems of classification. *Computación y Sistemas*, 19(1):109–118, 2015.

[35] Standard guide for recognition and derecognition of environmental liabilities. http://www.astm.org/cgi-bin/resolver.cgi?E3123-18. ASTM International - Standards Worldwide.

[36] Astm f2926 - 18. http://www.astm.org/Standards/F2926.htm. ASTM International-Standards Worldwide.

[37] WFL Bogaerts. Stp1257. http://www.astm.org/DIGITAL_LIBRARY/STP/PAGES/STP15415S.htm. ASTM International-Standards Worldwide.

[38] David G Lowe et al. Object recognition from local scale-invariant features. In *iccv*, volume 99, pages 1150–1157, 1999.

[39] Shree K Nayar, Sameer A Nene, and Hiroshi Murase. Real-time 100 object recognition system. In *Proceedings of IEEE international conference on robotics and automation*, volume 3, pages 2321–2325. IEEE, 1996.

[40] Edward Smith. Re-engineering a trash/recycling collection vehicle based on challenging customer requirements. 2010.

[41] P Suja, Shikha Tripathi, et al. Real-time emotion recognition from facial images using raspberry pi ii. In *2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 666–670. IEEE, 2016.