Specialization of an Existing Image Recognition Service Using a Neural Network

OSKAR DAHL
SARA ERSSON
Abstract

To help combat the environmental impacts caused by humans this project is about investigating one way to simplify the waste management process. The idea is to use image recognition to identify what material the recyclable object is made of. A large data set containing labeled images of trash, called Trashnet, was analyzed using Google Cloud Vision. Since this API is not written for material detection specifically, a feed forward neural network was created using Tensorflow and trained with the output from Google Cloud Vision. Thus, the network learned how different word combinations from Google Cloud Vision implicated one of five different materials; glass, plastic, paper, metal and combustible waste. The network checked for 518 unique words in the input and ran them through two hidden layers with a size of 1000 nodes each, before having a one hot output layer. This neural network received an accuracy of around 60%, which beat Google Cloud Vision’s meager accuracy of around 30%. An application, with which the user can take pictures of the object he or she would like to recycle, could be developed with an educational purpose to let its user know what material the waste is made of, and with this information be able to throw the waste in the right bin.

Keywords

Neural networks, Machine learning, Tensorflow, Google Cloud Vision, Image recognition
Abstract


Nyckelord

Neurala nätverk, Maskininlärning, Tensorflow, Google Cloud Vision, Bildigenkänning
# Contents

1 Introduction ........................................... 1  
   1.1 Background ......................................... 1  
   1.2 Problem ............................................ 1  
   1.3 Purpose ............................................. 1  
   1.4 Goal ................................................. 1  
   1.5 Methods ............................................. 2  
   1.6 Delimitations ....................................... 3  
   1.7 Outline .............................................. 3  

2 Background ............................................ 4  
   2.1 Neural Networks ..................................... 4  
   2.2 Image Recognition ................................... 8  
   2.3 Google Cloud Vision API ............................ 9  
   2.4 TensorFlow .......................................... 10  

3 Method ............................................... 11  
   3.1 Methodologies ...................................... 11  
   3.2 The Process ......................................... 11  

4 Result .................................................. 15  
   4.1 Google Cloud Vision ................................ 15  
   4.2 Feed Forward Neural Network ...................... 17  

5 Conclusions .......................................... 19  
   5.1 Discussion .......................................... 19  
   5.2 Conclusion .......................................... 20  
   5.3 Future work .......................................... 20
1 Introduction

This section will introduce the subject area of this thesis work, formulate the underlying problem and state the purpose and goals with this report. Social benefits, ethics and sustainability will also be discussed, along with what methods that have been used and what delimitations there are. Finally, the outline of the report will be presented.

1.1 Background

A phenomenon which still is a problem in today’s society is that many people do not recycle their waste. The ones that do however, might not always know where to recycle it. If materials are thrown in the wrong containers the purpose of recycling becomes ruined, which makes it important to explore new ways to simplify a certain step in the recycling process - the sorting [13]. One idea is to use image recognition, which refers to the ability of software to recognize different objects in images, to identify which material the object is made of.

Interpreting the surroundings using our vision is an easy task for the average person. There is no need for spending time on going through what has been seen to be able to tell whether it is a person, an animal, a building, a writing or a place, we just know. While human and animal brains make vision and recognition of objects seem easy image recognition is a much more difficult task for computers [5].

1.2 Problem

To create a fairly decent image recognition program both a lot of time and knowledge is going to be needed. If the developer does not have the time, nor the experience within this area, one solution could be to use a pre-existing image recognition program [19]. However, this program might not be specialized for the needs of the product. How well can it be used to recognize different materials? How much can it be improved by using other factors, such as what type of object it is and what company produced the product?

1.3 Purpose

The purpose of this work is to simplify the process of waste management on an individual level by creating an easy method to determine what should be thrown where.

1.4 Goal

In this section goals from different aspects will be presented, such as goals with the project, what social benefits there might be with spe-
cializing image recognition and if there are any ethical or sustainable goals with it.

**Goal with the thesis work**

The main goal of this project is to test the capabilities of using a feed forward neural network to handle the output from Google Cloud Vision for more specific uses. This will be used in this project to handle waste management for five different categories; glass, plastics, paper, metal and combustible waste.

**Social Benefits, Ethics and Sustainability**

Being able to detect different materials in images can be beneficial within many areas. One example could be within recycling. If a person wants to recycle an object, but does not know what material the object is made of, this question might be answered using an image recognition program specialized for this exact task. If detection of different materials can prevent people from throwing non-combustible materials and toxic waste in the bins for combustible materials, the environment would benefit from this, since the energy waste and risks of contamination would be reduced [18]. Concerning ethics, a problem might occur regarding how the images are handled and saved, since this is a matter of integrity. Another factor to this problem is how Google stores the images and if they could be used for other purposes. If this would be the case, the question is how they would do it.

**1.5 Methods**

This project was divided into two different parts; a literature study to widen our knowledge within machine learning, the usage of Google Cloud Vision and Tensorflow, and a design and implementation part.

**Literature Study**

During the literature study information was collected about on what level a specialized image recognition program can identify certain materials and how well an open API can identify them, to find the needs and the viability of the project. To be able to create a neural network using Tensorflow, information about how neural networks are constructed and how they work was studied. Finally, to better understand the limitations of Google Cloud Vision and how to push these limits, information about Google Cloud Vision was collected.

**Design and Implementation**

This part of the project was followed through using an engineering-based method with multiple iterations, each with a prototype and
evaluation. During the initial iteration a first prototype was produced based on the knowledge collected from the literature study, while the following iterations were based on the previous tests with respect to accuracy. This process was repeated until the results were sufficiently accurate, or until the improvements had stopped. To be able to evaluate the prototype, the data, consisting of images of different materials within waste management, was divided into three different sets. The first set contained 60% of the data and was used for training. The second set contained 20% of the data and was used for validating each epoch in the training process. The third set contained the last 20% of the data and was used for testing the final trained network.

1.6 Delimitations

The image recognition service which will be used within this project is Google Cloud Vision. There are other similar services which could be used, but it was decided to exclude them simply because a choice had to be made and there was already some experience with the service within the project group. A feed forward neural network will also be used to handle the output from the image recognition service. Since we will not evaluate and compare different machine learning techniques with each other, the usage of a neural network might not be the method that provides the best results for this project.

There are a lot of different materials within waste management, but to draw the limit somewhere it was decided to focus on five different material categories; plastic, paper, metal, glass, and trash. The output material trash works as a miscellaneous category, catching the materials which should be counted as combustible waste.

The quality of the images is not a factor that will be focused on in this project, since it is something that affects the performance of Google Cloud Vision rather than the neural network itself.

1.7 Outline

In chapter 2 neural networks and image recognition will be described, which is the theoretical background to this thesis work. The underlying methodologies of the project will be presented in chapter 3, followed by a presentation of the results in chapter 4. Finally, there will be a retrospective in chapter 5 where the results are discussed and a conclusion is drawn, followed by an analysis of possible future work.
2 Background

If a person is shown a hand-written number, it is quite easy for this person to tell what number it is, even if it is somewhat poorly written. Furthermore, think about all of the different ways a specific number can be written regarding position, angle, style, thickness and size, to mention a few. How is it possible to write a program that can recognize hand-written numbers when there are an endless amount of ways to write each and every one of them? How can even larger problems, like making a car drive autonomously or interpret spoken language, be possible?

To be able to answer these questions we are going to look into how this problem is handled by the biological brain and how artificial neural networks mimic its behavior in section 2.1. In 2.2 we will present image recognition, explain how neural networks can be used within this area and what problems can occur within material recognition. Finally in section 2.3 we will write about Google Cloud Vision API and its advantages and disadvantages.

2.1 Neural Networks

This section will cover how artificial neural networks replicate biological neural networks, and how backpropagation and deep learning work and are used.

Biological Neural Networks

The human brain consists of a huge amount of neurons, which are all interconnected at points called synapses. Due to the high amount of input that each neuron gets from other neurons, and all of them working in parallel, the brain and its biological neural network is super complex.

The neural system can be divided into three different parts; the receptors, the neural network and the effectors. The receptors receive stimuli either from the outside world or internally, the information is then passed into the neurons as electrical impulses. The neurons process the information and make decisions upon it, and the effectors translate these electrical impulses to output, which will be sent to the outside world again [6]. Examining closer how the neurons communicate, neurons receive signals from other neurons through the dendrites. Each dendrite has a synaptic, weight, which gets multiplied with the incoming signal and processed within the cell body. Whether the outgoing signal is going to be passed along or not is decided by the activation function, which works as a threshold. What happens when humans learn something new is that the values of some synaptic weights and thresholds are changed. This way the brain can make different decisions based on previous experiences [17].
Artificial Neural Networks

To enable computers to solve similar complex problems, artificial neural networks are used as a way of replicating how humans learn. These brain-inspired systems are one of the main tools used within machine learning since they can recognize complex patterns in input data that would otherwise be too complex or too numerous for a programmer to teach the machine to recognize [12].

Introduction to Layers

Generally, neural networks consist of different types of layers; input, output and hidden layers, as shown in figure 1. The input layer, as the name suggests, handles the input. The hidden layers exist to handle the computations and to give the network its depth and complexity. Each layer consists of an amount of so called neurons, which each holds a number between 0 and 1. This number is called the neuron’s activation. Each neuron in one layer is connected to every neuron in the next layer, and each connection is associated with a certain weight [14]. The weight represents the strength of the connection. Each neuron also has a bias value; an indication of whether it tends to be active or inactive. Whether the neuron is active or not can be compared to the whether the signal is passed along or not in the biological network [17].

![Figure 1: An illustration of how the different layers are connected.](image)

The input layer consists of partitioned input data. To give an example of what input data can look like, a suggested partitioning could be that each neuron in the input layer represents a pixel, if the data is an image. Between the input and the output layers there are hidden layers, which handle the calculations of the weights and the activation numbers, which in the end will provide a solution. Moving over to the output layer, this layer, as the name suggests, provides the solution to the problem. The solution is associated with a probability, which is what has been calculated through every layer of the network [14].
How a Neural Network Works

As introduced previously, in every layer except the input layer, each neuron is connected to every neuron in the previous layer. The activation of a neuron is calculated using the activation of each neuron in the previous layer, the associated weights of each connection and the neuron’s bias as parameters in a certain activation function [3].

An activation function which is often used is the Sigmoid function, which takes any real-valued number and squashes it into a range between 0 and 1. This is especially useful in the output layer when the goal is to provide a solution associated with a probability. Another activation function that is even more often used is the ReLU (rectified linear unit). The reason for its popularity is that it is used in the majority of the convolutional neural networks. The ReLU is rectified from the bottom half since it has the value zero whenever the input is zero or less, but equal to the input otherwise.

What is given as a parameter to the activation function is the sum of each activation number multiplied with the corresponding weight, and in turn adjusted with the bias [17]. Neurons with a greater activation will naturally have a larger effect on neurons in the next layers. Finally, the neuron which activation is the greatest, is the one the system has provided as the solution to the problem. This number also represents the probability of the solution being correct [3].

Learning Algorithms

Since the weights and biases of a neural network are all random from the start, its performance right after creation is rather poor, i.e. it often provides an answer that is wrong, and possibly also with a high probability. For a neural network to improve its performance it needs to be trained with data. For this purpose there are three different types of learning: supervised, unsupervised, and semi-supervised learning [8].

The most used one is the supervised learning. It works by letting an algorithm learn how the input is mapped to the output, using a labeled training data set and letting the algorithm iteratively make predictions upon it. Since the data is labeled the algorithm is aware of when the predictions are wrong, and the weights and biases can therefore be corrected. Going back and changing the values in the neural network is called backpropagation, which will be explained later in this section. The whole learning process continues until the network is able to make predictions upon new input data that it has never seen before, on an acceptable level [8].

Supervised learning problems can be divided into regression and classification problems. Classification problems are when the output variable is a category, whereas regression problems are when the output variable is a value.

During unsupervised learning the training data set is unlabeled. Unsupervised learning can be divided into clustering and associa-
tion problems. Clustering problems are problems where the goal is to discover the inherent groupings in the data, whereas association problems are about wanting to discover rules, describing large portions of the data. Semi-supervised learning is nothing but a mixture of the two previous learning algorithms [8].

How a Neural Network Learns

The cost function is a function that takes in all the activation numbers, weights and biases as parameters and returns a cost value; an indication of how well, or badly, the network predicts output. The first step is to calculate the difference between the current output and the targeted output for every single training example. The difference is then squared. This is repeated many times to obtain an average cost for each training example. Finally, the averages are summed up to obtain the final cost value of the network. The goal when training a neural network is to find a way to lower the cost value; in other words to minimize the cost function. The process of regressing in the network and changing values to reach this goal is called backpropagation, and can be explained by following quote:

Backpropagation is the algorithm for determining how a single training example would like to nudge the weights and biases, not just in terms of whether they should go up and down, but in terms of what relative proportions to those changes cause the most rapid decrease to the cost [4].

To explain this quote further an example can be that if the input is an image showing a hand-written ‘2’, the correct output would be 0 for every input except for number ‘2’, where the correct output would be 1, due to the output representing probability. Since a higher cost means a higher probability of making errors, logically the nudges to the current output values should be proportional to the costs [4].

There are different algorithms to minimize the cost. It is common to use the classic stochastic gradient descent algorithm to calculate the fastest way to minimize the cost. Recently however, the Adam optimization algorithm has become popular within computer vision and natural language processing. The Adam optimizing algorithm is an extension to the stochastic gradient descent algorithm and can be described as follows:

The method computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients [9].

Overfitting

The amount of data and its diversity that a neural network is trained with has a large impact on how well it performs on new input data.
If a network is not trained with enough data, or if the data is too general, it will perform poorly when predicting on new input data, because of the lack of complexity. This phenomenon is called underfitting. However, training the network with too much data makes the connections in the network very complex, and the network becomes adjusted to perfectly fit this exact data set. When making predictions on new input data the network will no longer be generalized enough to perform well. This phenomenon is called overfitting and is one of the most challenging phenomenons within machine learning [7].

2.2 Image Recognition

A human can identify what is in a picture without much effort, however it is a very complex task. How can a computer know the difference between a bee and a ’3’? This section will describe how a computer analyses a picture and present what problems can occur when identifying materials.

Basic Image Recognition

Even basic image recognition can be fairly complex and can only be used reliably on simpler tasks like identifying handwritten numbers. This is done by giving each neuron in the input layer a pixel of the image as input. The number of neurons in each layer depends on how many pixels the image consists of. If the image size consists of 28x28 pixels, the input layer needs 784 neurons. The amount of neurons in the hidden layers can vary independently of the image size. However, 16 neurons in each layer will suffice in this example, and since the range of numbers goes from 0 to 9, the output layer consists of 10 neurons. This network will have a fairly high success rate of about 96% with some training, and will have some expected and unexpected problems. One of the problems is that it will always choose a number even if there are no numbers in the picture [3]. A second problem, which however is not obvious at this image size, is the number of connections between all the neurons. In our small network for a small image with a small amounts of output choices there are 12960 connections. Each one of these connections provides a value to the next layer and together with the biases the neurons have, the network contains 13002 parameters. When the network is scaled to handle larger images with more complex outputs, the number of parameters will rise drastically to unfathomable amounts and the network will most likely be overfitted [19].

Image Recognition for Materials

There has been a lot of research in image recognition, most of it has been in object recognition. Identifying different materials comes with its own set of problems. The biggest one is the fact that different materials can have similar properties such as texture appearance,
reflectiveness and object class. An example of this is translucency that can be both glass, plastic, wax, and more. Another example is a car, it can be a working car made out of metal or a toy car made out of wood or plastic.

The network needs to take all these properties into account and to do this it needs to extract different information for different properties. This can be the edges and curvature to recognize what object it is and more basic features as color and texture. However, to have an effective material recognizer, the program also needs to analyze the micro texture which is derived from the residual image. This image is everything that the bilateral filter removes, which leaves only the smallest details [16].

### 2.3 Google Cloud Vision API

When asked how machine learning is used inside Google, the response was as follows:

> Machine learning has been a cornerstone of Google’s internal systems for years, primarily because of our need to automate data-driven systems on a massive scale. This experience has provided unique insight into the right frameworks, techniques, infrastructure, and data that can help customers complete a successful journey toward getting value out of machine learning [1].

Thus, Google uses machine learning within the majority of their products, for example within Google Cloud Vision [1].

#### Advantages with Google Cloud Vision

The Google Cloud Vision API has many features. It can detect labels and logos, detect and extract text in a large set of different languages, detect general attributes in images and detect famous landmarks. The API can also detect explicit content, like adult or violent content, and detect multiple faces in images. It can also search the Internet for similar images. Another advantage is that there is an integrated REST API which provides access to request annotation types when uploading images [11].

#### Disadvantages with Google Cloud Vision

Although the API supports facial detection, it does not however support facial recognition. This means that the API can detect faces in images, but not recognize who are in the images such as if they are the same person or if they are different people [11]. Google Cloud Vision does not provide video analysis, which could be seen as a disadvantage. However, Google has another API for this exact task, called Google Cloud Video Intelligence [10].
2.4 TensorFlow

The groups of people that use machine learning are most commonly researchers, data scientists and developers. To facilitate these groups working together TensorFlow was created. TensorFlow is an open source software library for dataflow programming, or high performance numerical computation, and can be run on both multiple CPUs, GPUs and TPUs, and on mobile operating systems like Android and iOS. The library provides APIs for several languages, such as Python, C, C++, C#, Haskell, Java, Go, Rust, Julia, R, Scala and OCaml [2].
3 Method

In this section the project’s methodologies and the whole process will be described.

3.1 Methodologies

Since the philosophical assumption is the starting point of any research we started looking into what would be ours by reading about each assumption. We derived that we were to apply a realistic assumption, seeing as it assumes that the reality is known and does not depend any people’s thought. Also we were going to work with understanding the collected data and using our knowledge to develop even better solutions, which made realism a good fit.

Knowing that our research would involve solving a known problem related to a particular situation, the applied research method was chosen for the research. The argument was that the known problem concerned detection of different materials, and that the particular situation was recycling. The fact that we were going to use a completed API also played a big part in choosing the most suitable research method, since the applied research method often builds on existing research and solving the problem with the help of data from the actual work, to develop applications.

Concerning the approach with which the research was going to be applied, it leaned more towards an abductive approach. This is because when drawing conclusions we did not start out with a general statement or hypothesis that needed to be proved, nor did we try to draw conclusions based on a lot of specific observations [15].

3.2 The Process

The idea was to let Google Cloud Vision analyze images and use the received word combinations and the corresponding labeled material to train the neural network to be able to identify materials in new images.

In this section the entire process will be described starting with how to find and extract the data, followed by how to prepare the input for the neural network and how to create the neural network. The section ends with explaining how the neural network was trained, validated after each epoch and tested.

Finding and Extracting Data

The implementation phase started out by searching for a large enough image data set that would be sufficient to train the neural network. The idea was to make the network able to guess which material the object in each image is made of. The data set had to contain images of different recyclable materials, and each image should contain only one single object to keep the determination of material as simple and
straight forward as possible. The object could however be made of several materials, so that the network would be trained to recognize materials in more realistic images.

An image data set, called Trashnet, was found on Github which was free for anyone to use, containing around 2500 images of garbage divided into following categories: cardboard, glass, plastics, paper, metal and combustible waste. Trashnet was created for a school project and the neural network created using this image set received an accuracy of 75%. It was decided not to use the cardboard folder, since it was not included in the project plan from the start.

When the data set was decided upon a project at Google Cloud Platform was created and all of the images were uploaded in the project’s bucket at GCP. A bash script was written which, by using Google Cloud Vision’s REST API, sent each photo from the bucket to Google Cloud Vision and saved the response as a json file locally, as shown in figure 2.

![An example metal image.](image1.png)

![Corresponding json file.](image2.png)

Figure 2: Comparison of an image and its response from Google Cloud Vision.

Another bash script was written to gather all the responses in one single file. The script did this by converting all the json files to a format that would be simple to parse when reading the data, and then adding each conversion as a new line to one single csv file called all data. This was done one category at a time.
The first row in figure 3 represents an image labeled as *trash*, which had received the words *crystal* and *plastic* via the REST API. Each row consists of the words from Google Cloud Vision, separated with commas, and ends with the index of the labeled material. Since there were five different materials each material was mapped to an index between one and five. The amount of words from an image was limited to 20, simply because no image in the data set ever received more than that amount in the response json file. For each image, when there were no more words, the script filled out the rest of the spots with spaces to simplify the parsing of the data later on.

### Preparing Input for the Neural Network

The next step was to extract the words from the *all data* file. This was done running a python script which split the words on commas. Every word that was not a blank space was added to a word list, which later was wiped of doublets by turning the list to a set, and then turning it back to a list again.

To be able to feed these words to the neural network in a way they could be interpreted each word was mapped to an index. The word list was sorted to assure that the words would receive the same index independently of how the mapping array was created. Since there were 518 unique words in total, a space matrix with the size of 518 for each image was created. Each array was obtained by going through every word from an image and putting a ‘1’ in its corresponding index in the image’s space matrix. The image input arrays were stored by creating a two-dimensional array and adding all of them one at a time.

Finally, the *all data* file was split into three different files; one for training which was 60% of the images, one for validating which was 20%, of them, and one for testing, which was the last 20% of the images.

### Creation of the Neural Network

The neural network was written in python using the open source machine learning framework Tensorflow. The three different data files were read and handled in the same way, they were however handled in a specific order which will be explained under *Training, Validation and Testing*. Each file was read as a csv file and saved
as 21 columns; 20 for the words and one for the label. The word columns were then saved as a matrix and sent to the input layer, and similarly, the label column was saved as a matrix and sent to the output layer.

The size of the input layer, how many hidden layers there were going to be and the size of the output layer was defined. The next step was to give each hidden layer and the output layer random weights and biases. Between each layer computations were defined using matrix multiplication, addition and an activation function called ReLU. The input to each computation session was the previous hidden layer, with all of its weights and biases. After that the loss was defined and used when using the optimizing method AdamOptimizer to be able to minimize loss.

**Training, Validating and Testing**

The training set was run through the neural network during 250 epochs. After each epoch the network was validated by running the validation set. Whenever the accuracy of the validation was higher than the previous highest rate, the state of the network was saved to preserve the best version of the network for the final outcome of the training session. When the training session was over the version of the network with the highest accuracy was tested one last time by running the test set. This accuracy was then counted as the final accuracy of the entire neural network.
4 Result

In the result section firstly the results using only Google Cloud Vision for material detection will be presented. Secondly the results from the trained neural network will be presented.

4.1 Google Cloud Vision

When only using Google Cloud Vision for material recognition, for it to count as a hit at least one of the words in the response had to include the label of the image. An example is figure 4, which shows an image labeled as glass, that after having been run by Google Cloud Vision received glass as one of the words in the response. The prediction on the image was therefore counted as a hit. A prediction on an image counted as a hit if and only if the material label somehow existed in the response, such as glass bottle or just glass, if the label was glass.

Figure 4: An image labeled as glass, which received glass in the response from Google Cloud Vision.

If the response contained several of the material labels it was counted as a guess for all of the mentioned materials. The images that did not have any of the material labels in their response were counted as non-labeled, such as the image in figure 5, which response only consisted of the words product and hardware.
In the image data set there were some images that were a lot harder for Google Cloud Vision to predict on, than the images in figure 4 and 5. An example is the image in figure 6, where the main visible item is a lid made out of metal because of the angle the picture is taken from, but is labeled as glass since the main object is a glass jar. This image received the words brass, metal, and silver.

Google Cloud Vision had an overall accuracy of around 30% when
running the entire data set. In the left-most column in table 1 are the actual materials the objects in the images are made of. In the top row in table 1 are the materials guessed by Google Cloud Vision. The right-most column in table 1:results represents the share of photos which Google Cloud Vision did not have a material guess on.

Table 1: Results in percent from Google Cloud Vision used as a material recognition API.

<table>
<thead>
<tr>
<th>Actual materials</th>
<th>Glass</th>
<th>Plastics</th>
<th>Paper</th>
<th>Metal</th>
<th>Trash</th>
<th>Non-labeled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glass</td>
<td>61.89</td>
<td>5.53</td>
<td>0</td>
<td>1.23</td>
<td>0</td>
<td>31.35</td>
</tr>
<tr>
<td>Plastics</td>
<td>0</td>
<td>37.9</td>
<td>0</td>
<td>4.93</td>
<td>0</td>
<td>57.17</td>
</tr>
<tr>
<td>Paper</td>
<td>0.17</td>
<td>2.62</td>
<td>8.74</td>
<td>0.52</td>
<td>0</td>
<td>87.95</td>
</tr>
<tr>
<td>Metal</td>
<td>3.51</td>
<td>1.51</td>
<td>0</td>
<td>22.56</td>
<td>0.5</td>
<td>71.92</td>
</tr>
<tr>
<td>Trash</td>
<td>1.58</td>
<td>9.52</td>
<td>3.97</td>
<td>1.58</td>
<td>0</td>
<td>83.35</td>
</tr>
</tbody>
</table>

4.2 Feed Forward Neural Network

To summarize, the neural network takes in a space matrix with the size of 518 as input, which represents what words Google Cloud Vision has included in its response for one image. The network then outputs one of the five different materials. The network was trained for 250 epochs using the training image data set. The prediction (the output material) the network made on each image counted as a hit if and only if it matched the image’s label. The training resulted in an accuracy, as seen in table 2, of around 60% when testing the neural network with the testing image data set.

Table 2: The results for the network

<table>
<thead>
<tr>
<th>Material</th>
<th>Glass</th>
<th>Plastics</th>
<th>Paper</th>
<th>Metal</th>
<th>Trash</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>52%</td>
<td>73%</td>
<td>78%</td>
<td>52%</td>
<td>8%</td>
<td>60%</td>
</tr>
</tbody>
</table>

When looking at both table 1 and table 2 it easy to see that the accuracy of this neural network is overall 30 percentages higher than Google Vision’s. It also performs better within all material categories, except for glass. It has the best accuracy on paper and the worst accuracy on images of so called trash.

The network predicts right on the image in figure 4. It does not predict right on the images in figure 5, nor on figure 6.
Figure 7: A more difficult image to predict on, labeled as *metal*.

The image in figure 7 above received the words *red*, *product* and *font* in the response from Google Cloud Vision, which is not anywhere near one of the five materials. Still the network was able to predict the right material - *metal*. 
5 Conclusions

This section will cover the analysis of the results of the project, discussing delimitations, choices of methods etc. Finally it will present the conclusion of the results.

5.1 Discussion

There are a few factors that could have effected the results from the neural network negatively. Ideally the data set should have been much larger to cover a wider range of garbage, especially considering that it detects information such as logos, since it is impossible to cover garbage from the majority of the products in the world with only around 2000 images. It also would have been better to have several photos of each product to somewhat cover the fact that an object can appear in photos in many different angles, and that garbage usually is not handled carefully and easily becomes worn and torn. It was however difficult to find the perfectly combined image data set, partly because of the time limit of this project, and partly because it is difficult to even know what a perfect image data set is in the first place.

Even though the images for the training, testing, and validation sets were divided randomly there is a risk that the data sets were too similar, which means that the network potentially could be overfitted in disguise. Similarly, if the sets were too different the network could be underfitted in disguise instead. Both of these phenomenons would worsen the performance of the neural network when predicting on new images [7].

Another contributer to a possibly poor accuracy might be the comparatively small amount of words in the input layer. The network is only trained with words from Google Cloud Vision when running the images from Trashnet, which could have a negative effect on its performance on new images if the data set lacked variety or variety of products in general. In the real world there certainly are more words than the unique ones Google Cloud Vision gave as output. Therefore the real-world usage of this neural network probably has a worse accuracy than what the test accuracy suggests. However, if there are words that only occur in the test and validation sets their weights and biases will be unchanged and behave more like a real-world photo.

Another problem could be the loose term trash, since Trashnet’s definition of trash might not be the same as Google Cloud Vision’s definition. This might explain why Google Cloud Vision did not have a single correct answer when predicting on the images in the within the trash category.

As seen in figure 6 in the result section, there is always a risk that Google Cloud Vision returns misleading words. These words will most-likely result in a faulty prediction from the neural network,
unless there were many images of the same kind, or with similar contributing factors. Even a human would think that the words brass, metal, and silver, which was the response when Google Cloud Vision predicted on the image in figure 6, describes a product made out of metal of some kind.

5.2 Conclusion

Even though the accuracy of our network is not super high, at least it is twice as high as the accuracy when only using Google Cloud Vision for material recognition.

Our simple model can be compared to the one by the creators of Trashnet. Their network had an accuracy of 75%, which is higher than our, but at the same time they used a more advanced technique which included both a convolutional neural network and a support vector machine.

Because of an accuracy higher than Google’s and the limited knowledge needed, potential has been seen using a machine learning method to handle the results from an already complete image recognition API. Even though the chosen method might not have been the best one, the knowledge needed to create a material recognition program or API from scratch would have been much harder, time consuming and way out of scope for this project.

Using a neural network to specialize an already existing image recognition API on recognizing materials can greatly improve the performance.

5.3 Future work

A feed forward neural network might not be the best way to solve this problem, and future work could explore other options to solve it, such as a support vector machine. It might also be interesting to see what the results would be using a much larger data set to train the network on, or if another image recognition program or API was used, and compare the results.

Using image recognition to determine what material on object is made of could be used to sort waste automatically in a more efficient way, or detect garbage that has been thrown or sorted in the wrong bin.
References


