Engineering's Degree Project

Separation and extraction of valuable information from digital receipts using Google Cloud Vision OCR.

Author: Elias Johansson
Supervisors:
Johan Hagelbäck,
Ragnar Martinsson
Semester: VT 2019
Subject: Computer Science

Abstract

Automatization is a desirable feature in many business areas. Manually extracting information from a physical object such as a receipt is something that can be automated to save resources for a company or a private person. In this paper the process will be described of combining an already existing OCR engine with a developed python script to achieve data extraction of valuable information from a digital image of a receipt. Values such as VAT, VAT%, date, total-, gross-, and net-cost; will be considered as valuable information. This is a feature that has already been implemented in existing applications. However, the company that I have done this project for are interested in creating their own version. This project is an experiment to see if it is possible to implement such an application using restricted resources. To develop a program that can extract the information mentioned above. In this paper you will be guided though the process of the development of the program. As well as indulging in the mindset, findings and the steps taken to overcome the problems encountered along the way. The program achieved a success rate of 86.6% in extracting the *most valuable information*: total cost, VAT% and date from a set of 53 receipts originated from 34 separate establishments.

Keywords: optical character recognition, automatic text extraction, python, google cloud vision, string analysis, receipts.

Contents

1 Introduction

Ineffective administrative hackwork is something that can be found regardless of professional sector. The necessity of transferring information from a physical object like a receipt or document to a database can be daunting. The time and effort spent on performing this task could instead have been invested in something more productive. I believe that there are many areas within an organization that can be considerably more effective if we were able to alleviate the necessity for an employee to manually transferring data by implementing an Optical Character Recognition (OCR) with a tailored software that could automate the process.

1.1 Background

1.1.1 OCR

OCR is an abbreviation for Optical Character Recognition and it is a technique intended to identify and recognize text within an image. The idea closely resembles the process in which we humans recognize text within different environments. We visually detect text in our surroundings by processing visual information from our eyes and detect contrasts between colours. The contrast between the colours will be represented by shapes which we interpret as the characters, number and other symbols. This procedure is commonly used to digitize a hard copy of a document by extracting the text using a software, which alleviates the necessity of manually transferring the information [1]. However, the technique can be implemented in most areas regardless of organizational sector. With OCR, organizations are able to store their physically stored text data and numerical data by digitizing it. Which in turn will increase the integrity of the information by reducing the risk of harm coming to the physical data such as fire, pests or general degradation by time. Digitizing of data will also provide remote access to previous inaccessible data [2].

In this project this technique will be used to extract text from a digital image of a receipt.

1.1.2 Pre-processing of images

To increase the accuracy of the OCR program the image may have to be altered using pre-processing to increase the contrast of the text and the surrounding environment. The processing alternatives that will be tried in this project are:

a. Grayscale the image to increase the contrast between the background and the actual text.
b. Resizing of the image to achieve better result in detecting individual characters or increase/decrease the separation of words.

1.1.3 Post-processing of data

The result of the OCR algorithm may have inaccurate result in terms of interpreting the character from the text within the image. In some cases when the quality of the image is inadequate the OCR will find it difficult to differentiate characters such as:

- ‘x’ from ‘%’
- ‘0’ (zero) from ‘O’ or ‘o’
- ‘I’ (capital i) from ‘l’ (lowercase L)

In these cases, the result of the interpreted word may differentiate from its original. Since a Python script will be implemented to process the result of the OCR it is imperative that in the case of slight alteration of the text the script will still be able to determine the probability that a misinterpreted word correlate to another. This will be done using a ratio method in the script that will be explained in more detail further down in the text.

1.1.4 Google Cloud Vision API

Google Cloud Vision [3] is an API (Application Programming Interface) that provides a set of tools that can be used to perform image analysis. The API can either be used with their pretrained models or it can be used to build custom models using AutoML Vision. In this project the prebuild features will be used exclusively. Google Cloud Vision provides a set of tools that can be used to perform a wide variety of operations such as; label detection, landmark detection, web detection, face detection, content moderation, ML Kit integration, handwriting recognition, product search, image attributes, object localizer, logo detection, integrated REST API, and optical character recognition.

1.2 Related work

Text extraction from an image relies on many previously implemented methods. The accuracy of the text extraction correlates to the quality of the image as well as the condition of the object the image depicts. Pre-processing of an image may increase the accuracy of the extraction.

In [4] Bieniecki, Grabowski and Rozenberg; describes a set of pre-processing steps to implement in order to increase the conditions for the OCR using images from a digital camera. The OCR engine that they used in order to validate the pre-processing steps was an OCR-engine called FineReader. Even though their OCR-engine differ from the one that have been used in my paper; the pre-processing steps may still apply.

Pooja and Shanu describes countermeasures to implement in order to combat quality- and information-degradation of digital images of text documents. Quality enhancement techniques may have to be implemented to reduce the noise of the image in order to improve the extraction capabilities of an OCR engine. In the paper [5] they evaluate the performance of different pre-processing techniques applied on camera captured documents.

Yasser Almodhi [6] conducted a bachelor thesis project where he classified receipts and invoices using Machine Learning. In that project Yasser implemented methods related to this project such as; pre-processing of images as well as text extraction of text using Optical Character Recognition.

1.3 Problem formulation

Extracting information from a digital image is not something new or infeasible to implement. There is a magnitude of different software on the market that is designed to do just that. Extracted information can be considered to hold varying amount of value depending on the main goal of the extraction. In the case of this report where the main goal is to extract information from a digital image of a receipt; some parts of the information may not be equally valued as other parts. In order to extract valuable information, we will first have to determine what information of the receipts that we actually want to extract and what parts that is determined insignificant enough to neglect.

The main goal of this project is*:*

To create a script that can extract information from a set of receipts. Using an already existing OCR-engine, Google Cloud Vision.

The information to be extracted can be divided into two main categories:

1. ***Valuable Information = Total cost, Value Added Tax (VAT), VAT%, Gross-cost, Net-cost and Date.***

2. ***Mission Values: Total cost, VAT% and Date.*** (These values are the most desirable to extract)

Given that different receipts may have a completely different appearance and structure it could be interestingly complex to create an algorithm that would extract the valuable information regardless of its location on the receipt.

Extracting information from an image in some cases might not be ideal; in order to perfect the solution, it is desirable to implement an algorithm that can extract the information with sufficient success rate. Parts of the information that will be extracted such as cost or tax rate is something that cannot be left to chance to be interpreted in the correct way.

A way to increase the success rate of the text extraction can be to implement pre-processing of the image to make the conditions more favourable for the OCR software. Another way would be to implement a post-processing step where the resulting text from the OCR may be altered to increase the success rate of the Python script. If some words in the text correlating to total cost, added value tax or tax rates has been misinterpreted some sort of spellcheck will be implemented in order to change the misinterpreted word into a more probable word.

1.4 Motivation

Automatization in our society as a whole can be a valuable asset to have. By relieving employees the necessity of performing mundane tasks by implementing software that can automate the process, the employees will be able to channel their focus and energy into something more productive. Additionally, automatization can, if implemented correctly, reduce the total amount of time necessary to perform a task. Optical Character Recognition (OCR) is a valuable factor in digitalization parts of our society and the areas of which it can benefit is seemingly endless.

One example of where OCR, digitization and automatization of processes can greatly benefit the organization is health care. By transferring information from a physical object such as medical reports, laboratory tests result or similar information by implementing an OCR with a correlating automatization software the management of patient data can be much more efficient. Additionally, by digitizing the information it could make it more accessible for hospital staff which in turn can yield a more precise and effective treatment. Even though the goal of my implementation will automate the process of extracting information from a receipt a similar thought process could be used to design any sort of information using OCR.

1.5 Objectives

Objectives	Description
O1	Determine what OCR/API that is suitable for the task.
O2	Set up necessary configurations and prerequisite to make the OCR/API able to work.
O3	Write a Python script to use the implemented OCR/API to extract data from a digital image.
O4	Determine if any pre-processing of the image is necessary to increase accuracy of the text extraction.
O5	Update the Python script to achieve enough extraction of valuable data for a single receipt.
O6	Implement a way to extract valuable information from a set of differently structured receipts.
O7	Evaluate the final solution. How accurate are the extraction of the mission values?

Table 1.1: Description of objective to be covered during the project.

1.5.1 Personal assumptions of the result

Since the time for this project is strictly limited I believe that the project will result in a program that can extract valuable information from a set of predefined receipts with a few different text structures with sufficient accuracy. Since my program relies on an existing Optical Character Recognition software, Google Cloud Vision OCR, the result of my program will rely strictly on the ability of the OCR to extract the text. With that said, I believe that this OCR is a strong program with enough text extraction capabilities to satisfy the needs of this project.

I assume that Google Cloud Vision OCR will come with configured set of pre-processing methods that the API uses to improve the text extraction. However, some pre-processing might be needed in order to maximize the capabilities of the OCR engine.

1.6 Scope/Limitation

The possibility of implementing a fully-featured and completely optimized application that will extract data from any receipt with full accuracy is way beyond the scope of this project. I will limit the project to be able to handle extraction of data from a set of receipts that represent the most common used receipts at the company which this project is carried out at, namely *HRM*. I have also decided not to implement any machine learning in this project, I will rely on manipulating strings and looking for substrings of text extracted by the OCR-engine. A few pre-processing methods will be implemented, to see if the accuracy of the text extraction can be increased. The end goal of this project is to create a script that can extract valuable information with good success rate from a set of receipts. The set of receipts will consist of 53 receipts originating from 34 separate establishments, collected by the employees of HRM. All data in forms of receipts will be of Swedish origin. The language of the receipts will also be restricted to receipts containing Swedish text.

1.7 Target group

OCR and other forms of image analysis can be implemented in any field of work or research areas. The primarily group interested in this project are individuals or organizations that want to create their own OCR application and want to see if it is

possible to extract valuable information from an image with a limited amount of resources using Google Cloud Vision OCR in combination with Python.

1.8 Outline

Chapter 2 - Method, in this section of the report I will explain what method I used to conduct the project.

Chapter 3 - Implementation, in this section I will explain the resulting program that I have developed during this project.

Chapter 4 - Result, in this section I will present the result; the accuracy of the program and the result of whether the pre-processing of the images in combination with the post-processing of the result can yield higher accuracy of the extraction of the text.

Chapter 5 - Analysis, in this section the result will be analysed. Along with what conclusion can be drawn from the result.

2 Method

To create the most suitable approach in how to extract the valuable information from the result of the OCR a set of controlled experiments will be conducted. Experiments and tests will be run continuously throughout the development of the program in order to create the most suitable program possible. Since different receipts have different structures and represent the valuable information in different ways we will have to perform the test of a wide variety of receipts to cover as many structures as possible. In this chapter the experiments on how to determine the structures of the receipts and how to extract the desired data will be presented.

2.1 Experiment Structure

The experiments on how to determine the structure of the receipts will consist of assessing a collection of receipts and proceed to take a picture of each receipt. The picture will then be processed with the Google Cloud Vision OCR where a request containing the picture is sent to the server that handles the requests and replies with a JSON structured text format containing the text within the image. The text extracted by the OCR-engine will be structured differently depending on the structure of the text within the image. The resulting structure of the extracted text may be drastically different from other extracted text. In order to collect as many structures as possible we will have to look at as many differently structured receipts as possible.

When a new text structure has been encountered in several receipts, the program is updated in order to extract the desired information from that particular text structure. It is important to note that the program must be updated in a way that makes it backwards compatible; if the program is updated, the functionality of extracting the earlier structures will also be possible. If a completely new structure is encountered that is not compatible with the precious implementation a new function may have to be implemented in order to cover the text extraction. Ideally, the existing text extraction function will be updated to handle both cases. The more differently structured receipts collected the more prominent and effective the program will be.

2.2 Reliability

To reproduce the same results as this project would be achieved if the same steps have been taken as described in the implementation chapter of the report as well as using the same infrastructure and the same version of the Google Cloud Vision OCR API. Note that as mentioned previously in the report; different receipts will be structured differently, which means that if the program will be applied on a receipt that the program has not been tried on previously the result may vary. Since the program relies on the result of OCR program a deviation of the expected result may be seen if the resolution and/or quality of the image alternates from the ones expressed in this project. Since this project relies on the ability of a user to contribute a picture of sufficient quality the result may also vary. With that in mind, this is only one way to extract the information from the text obtained by the OCR. A small alteration of the steps in the implementation of the program may most likely result in a variation of the result.

2.3 Validity

To validate the result of the project and its experiments it is important to control the result by running tests on a set of similar receipts in order to determine that the program can extract the valuable information with sufficient accuracy. The keywords *valuable information* and *sufficient accuracy* may be interpreted in different ways by different readers. In order to claim that the program can extract valuable information we will first have to define what that information is. In this report, as mentioned previously in the report, the information that will be considered valuable are: total cost, value added tax (VAT) and tax rates (VAT%), gross-, net-cost and date. To determine what sufficient accuracy is I will conclude that the program works as intended if it is possible to extract the valuable information most of the times given that the pictures are of high quality and the receipts are of high quality. If the program is able to extract: total cost, VAT% and date in around 90% of the receipts; it will be considered to have sufficient extraction capabilities.

Another factor that may cause variation of the result is the environment in which the picture is taken. It is important that all hidden factors such as illumination in the environment in which the picture is taken will not reduce the quality of the image. There is a magnitude of factors that may cause a variation of the result, so it is important to make sure that the tests are conducted in as similar environment with as similar factors as possible in order to achieve decent validity of the results.

3. Implementation

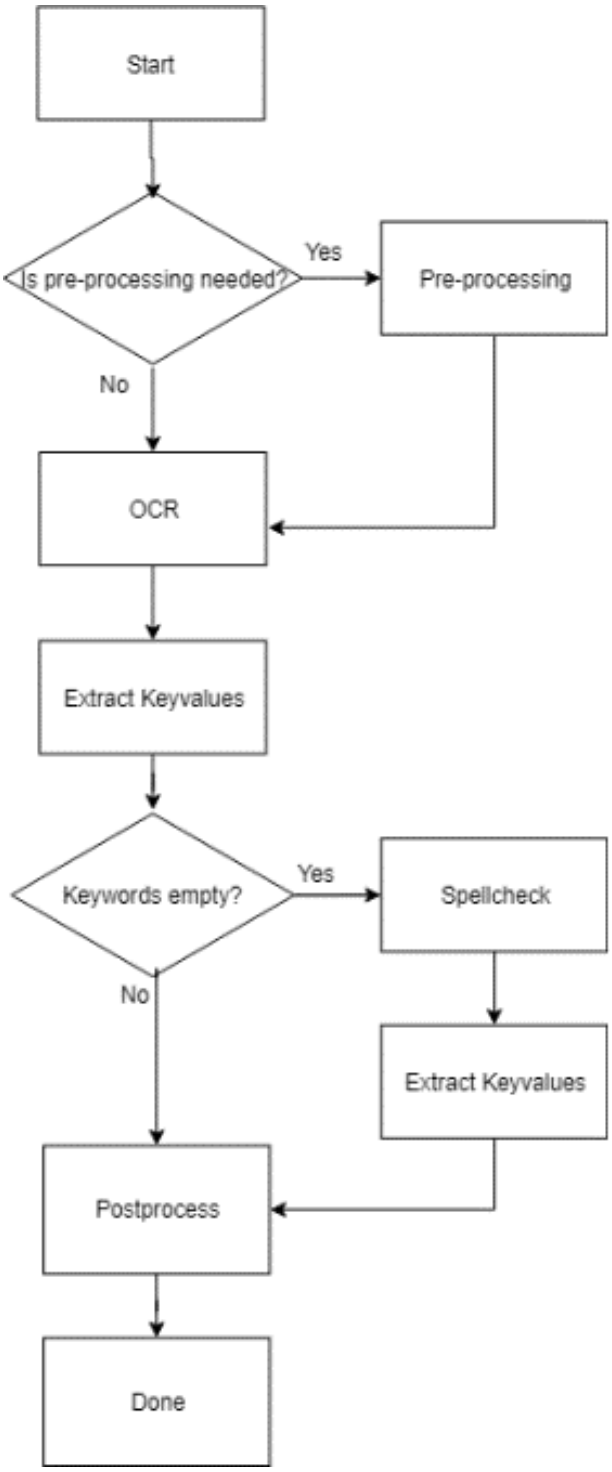

Figure 3.1: Visual representation of the steps taken of the program.

Figure 3.1, seen above, outlines the entire process used to extract the valuable information from a receipt. It starts by determining if any pre-processing of the image is required to extract the information; this is currently done manually by analysing the result of the text extraction. This part may be automated as a part of a future work of this project. If any pre-processing of the image is required a set of pre-processing

methods will be carried out. Each step after this section is automated: the text within the image of the receipt is extracted using the OCR-engine, Google Cloud Vision. The valuable information is then extracted from the result of the OCR-engine using the developed python script. If the script failed to extract some valuable information, a spellcheck will be run. When the spellcheck as been run the script will extract the valuable information again. Regardless of the result of the second extraction the script will enter the final stage; *postprocess*. In this stage the script will try to calculate any missing values using the information which was successfully extracted. In the following sections each individual step will be described in more detail.

3.1 Pre-Processing

Since the OCR engine, Google Cloud Vision OCR, will be used throughout this project all future steps of this project are based upon the ability of the OCR to extract the correct information. To ensure the best possible conditions for the OCR to work pre-processing of the images may be necessary. Two methods of pre-processing have been implemented; rescaling and grayscale of the image. A Python module called PILLOW [7] has been used to achieve the pre-processing step of this project.

The factor that contributes the most to how well the OCR-program can perform is the quality of the source image [8]. Some guidelines of how to ensure good quality of the image would be to ensure that the receipt of which the user intend to take a picture is not wrinkled nor damaged in order maintain the proper text alignment of the receipt. It is also important that the receipts do not contain any colour variation in form of stains which may reduce the contrast of the colours. Something that in turn may reduce the quality of the text extraction. It may also be important to note that the quality of the printed text has effect on the accuracy of the OCR-program. If the text is printed with a low-quality ink on a low-quality paper it may result in an illegible text.

Since receipts are a token of acknowledgement that a transaction has occurred between two parties [9]; the individual participant in the transaction may not be able to alter some of the attributes of the receipt once it has been received. Some of the attributes of the receipt that may be hard to alter once the receipt is received is the quality of the printed text. This leads to the possibility of pre-processing of the image to enhance the quality of the unchangeable attributes.

3.1.1 Resize

In some cases, resizing the image may enhance the accuracy of the text extraction of the OCR-program [8] [10]. DPI (Dots Per Inch) is a measure of the pixel per inch that a digital image can have [11] [12]. The measure can be seen as a ratio that denotes the resolution of the image. By increasing the DPI ratio, we can increase the representation of the information within the image with higher accuracy. By increasing the number of pixels used to represent the information of the image; the more precise the representation can be to the original object. By resizing the image, the quality of the image may not necessarily increase. Rescaling the image may give the OCR-program a higher chance to differentiate characters within a word of the source image. Increased accuracy can be achieved by either increasing or decreasing the size of the image; depending on the state of the source image. In this project, the size of the images are varied by a factor of: 0.5, 1.5 or 2

3.1.2 Grayscale

Grayscale or binarization of the image refers to the process of representing a colour image with white, grey and black colours [13]. By applying this method of pre-processing the contrast between the pixels representing the characters may be increased. Another advantage of applying a grayscale of the image before sending the picture to the OCR-program can be that the grayscale of the image may be of lower size than the source image. Something that is desirable in the case of Google Cloud Vision OCR since the request has a limitation of 10MB.

3.1.3 Comment

Pre-processing of the images has not been used in most cases for this project. Google Cloud Vision has achieved enough data extraction without the necessity of pre-processing. In the cases where the OCR-program failed to extract the correct data from an image; a new image of the same receipt was produced along with a new OCR request. That in most cases yielded a sufficiently accurate data extraction. Pre-processing was used for the images that required to be retaken. To determine whether pre-processing can be used to achieve better data extraction the images that failed to produce enough data extraction were put through the pre-processing steps mentioned above. The result of these test will be presented in the result section of this paper.

3.2 OCR

3.2.1 Google Cloud Vision OCR

This is the API used to extract the data from a digital image. A Google Cloud Project is required to use the functionality of the API. Information on how to set it up can be found on the following webpage [14].

Google Cloud Vision has a request-based text extraction method that will be used in this project. This approach will be restricted in term of the size of the request. The size of the requests may not exceed 10MB, which can be problematic given that the higher resolution the photo has the more bytes are required to represent the data. To combat this image used in this project will be of the JPG format. This is a lossy compression format [15] which reduces the bits required to represent the data of the image. The downside is that after decompressing the image some of the information may be lost in the process. Using this OCR and producing an image if JPG format the following result can be expected:

Figure 3.2: Picture of a receipt to be run though the OCR-engine.

```
la castellina
dagens
1 109.00 109.00
total t sek
kontokort fast
tillbaka
109.oo
109.00
0.00
# moms %
2 moms 12%
moms
11.68
netto
97.32
totalt
109.00
swedbank babs. butik: 2538023. term: 12003221
kep. 2019-03-25 12:52
sek: 109.00/moms: 11.68/totalt: 109.00
debit mastercard ix** i**
k/2 1  she 547 li 1528. ref.nr: 1200322 11 551
a1d:a0000000041010 tvr:0000000000 tsi:e800
spara kvttot kundens kopia
kontaktlös
xx 512
thaungs restaurang ab
liedbergsgatan 11. växjö
orgnr: 559007-7805
0470-701003
infoélacastellina.se
valkommen äter!
ni betjänades av: lunch
kontroll # etaxa 101104007041
353224 12:52:48 2019-03-25 37639 1 1
```

Figure 3.3: Result of text extraction from image (Figure 3.2) using Google Cloud Vision

3.3 Text Extraction

In order to structure the result of the text extraction in an understandable and cohesive manner the receipts will be grouped into a subset of receipts: *Lunch, Groceries, Travel, Gas and Other*. Note that the groups of the receipts are only used in presenting the result. These groups will not have anything to do about the logic of the script that will handle the text extraction.

3.3.1 Lunch

This set contains 16 receipts from 11 separate establishments.

Establishment	Number of receipts
La Castellina	5
Cafe Coretto	1
Hong Kong	1
King of India	1
Kungsgrillen	1
Abbas Falafel	1
Max	1
Monte Carlo	1
Pizzahut	2
Stars and Stripes	1
Rustique	1
Total establishments: 11	Total receipts: 16

Table 3.1: Number of receipts in this subset along with their corresponding establishment.

3.3.2 Groceries

This set contains 7 receipts from 3 separate establishments.

Establishment	Number of receipts
Ica Kvantum	4
Ica Supermarket	2
Willys	1
Total establishments: 3	Total receipts: 7

Table 3.2: Number of receipts in this subset along with their corresponding establishment.

3.3.3 Travel

This set contains 13 receipts from 6 separate establishments.

Establishment	Number of receipts
Arlanda	2
Kalmar Länstrafik	2
Länstrafiken Kronoberg	3
Region Kronoberg	2
Stockholms Lokaltrafik	1
Taxi Stockholm	1
Total establishments: 6	Total receipts: 13

Table 3.3: Number of receipts in this subset along with their corresponding establishment.

3.3.4 Gas

This set contains 4 receipts from 3 different establishments.

Establishment	Number of Receipts
OKQ8	1
Circle K	2
Preem	1
Total establishments: 3	Total receipts: 4

Table 3.4: Number of receipts in this subset along with their corresponding establishment.

3.3.5 Other

This set contains 13 receipts from 11 separate establishments.

Establishment	Number of Receipts
Astas Blommor	1
Biltema	2
Granngården	1
Kastebergs Gård	1
House of Carlander	1
AQ	1
MQ Växjö City	1
Pressbyrån	1
Systembolaget	1
Teknikmagasinet	1
Apoteket	2
Total establishments: 11	Total receipts: 13

Table 3.5: Number of receipts in this subset along with their corresponding establishment.

3.3.6 Summarizing receipts

As seen in the tables above the set of receipts was constructed of 53 receipts from 34 separate establishments. In order to determine what extraction method to be implemented in order to differentiate the valuable information from the total text obtained from the OCR; the text from each receipt has to be studied in detail. After studying the result of the OCR each result boiled down to one, or more, of three main structures.

3.3.7 Script

Description

A script was developed in order to differentiate the valuable information, (VAT, VAT%, gross-, net-, total-cost and date), from the rest of the text obtained by the OCR as seen above. The script will look for keywords within the text corresponding to the valuable information and the corresponding values on the receipt. The collection of receipts in this project consists of a subset of receipts collected from a wide range of establishments.

Extraction Methods

Given the information of the wide variety of establishments the set of receipts of the collection from the precious section; a broad estimation can be made. If a correlation can be found given the text structure of the result from the OCR, an estimation can be made,

that these structures are a general text structure. In the following section the three most commonly encountered text structures will be presented.

Three extraction methods have been developed in order to extract the valuable information and the corresponding values from the text. These work in a similar way: given a keyword the script will iterate over each word of the text until the keyword has been found; given that the OCR works as intended the values corresponding to the keyword will be found in close proximity to the keyword. In the set of receipts used in this project three frequently occurring text structures can be seen. These structures can be seen in the figures below.

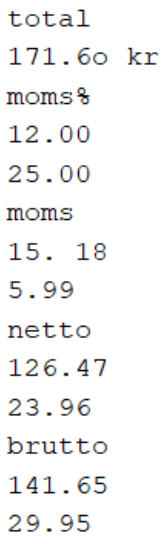

Figure 3.4: One of the three frequently encountered structures of the text extraction.

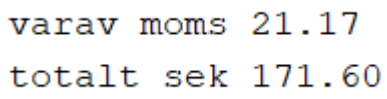

Figure 3.5: One of the three frequently encountered structures of the text extraction.

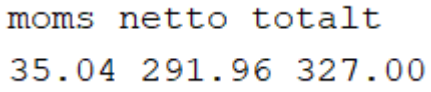

Figure 3.6: One of the three frequently encountered structures of the text extraction.

Most of the text structures found matched the structures seen above. Three text extraction methods have been developed to cover these cases.

Method 1

An extraction method has been developed to handle the first text structure seen in Figure 3.4. This method will look for keywords that are located on a separate line of text. When the script has found a keyword on a separate line it will enter a loop and examine the row beneath the keyword. If the next line is constructed of a numeric value; it will consider the value to be a potential match with the keyword. The value will be mapped to the corresponding keyword. The loop will break if the next line is not a numeric value. As seen in figure 3.4; when the script encounters the keyword: ‘moms%’ it will enter the loop and extract the values ‘12.00’ and ‘25.00’. When the loop encounters the line

constructed of the word ‘moms’ it will break the loop since ‘moms’ is not a numeric value and will most probably not be a potential value of the keyword ‘moms%’.

The program will continue to loop through the text and do the same set of instructions for each keyword found.

Method 2

This method will handle the extraction of data of the text structure seen in figure 3.5. This time the script will loop through the text and look for keywords that are within a line of text containing additional information. If a keyword has been found it will split the line to produce a set of words. The line will be split on the white-spaces dividing each word or value in order to differentiate them.

Given the following text:

Summa 471.70 28.30 500.00

The script will split the line on the white-spaces to create the following structure:
[‘Summa’, ‘471.70’, ‘28.30’, ‘500.00’]

The script will then loop through each element of the newly split line. If the line contains a numeric value, the value will be mapped to the keyword.

Method 3

The last method is to find multiple keywords found on the same line without any other information; as seen in figure 6. The script will in this case loop through the text until it finds a keyword. If a keyword has been found it will look for other keywords on the same line. If multiple keywords have been found it will enter a loop and examine the next line of text. If the lines below are constructed of numerical values, they will be mapped to the corresponding keywords.

Given the text structure seen in figure 6.

line_keyword = moms netto total
line_values = 35.04 291.96 327.00

The script will find the line containing the keywords; the script will then move to the next line containing the values. Each value on that line will be separated and examined individually. If it is a numeric value it will be mapped to the keyword on the same index as the value. Given the first value of the row: ‘line_values’.

first_element = 35.04

This is the first element of the line: line_values which will give it the index ‘0’. If this element is a numeric value it will map the value to the keyword corresponding to index ‘0’ of the line: line_keywords. The element corresponding to index ‘0’ of the line: line_keywords is the keyword: ‘moms’.

Given the two rows above the script will map the values as follows:

moms = 35.04
netto = 291.96
totalt = 327.00

What extraction method to use

Different extraction methods worked best for different receipt structures. To determine what extraction method to implement for what receipt is a difficult task to do. The way that the script is developed to worked is that each extraction method is used. In most cases only a few elements were extracted using each method. However, each extraction method is constructed in such a way that if a value was mapped to a keyword the same value cannot be mapped to the same keyword again.

Spellcheck

The result of the OCR is not flawless. Given the factors mentioned through this paper the OCR may misinterpret a character for another. This is something that will have to be dealt with in order to achieve a desired result. Since the script will look for substrings within a text in order to differentiate a keyword from the rest of the text; it is imperative that the keyword is not misspelled. This has led to part within the script that will work as a "spellcheck". The script will work in the following way; given a keyword the script will iterate over all the words within the text and look for words that are similar to the keyword. The similarity is calculated using a *ratio* of how similar the words are. The ratio is calculated given the characters of the words. In order to calculate the ratio a Python module called *SequanceMatcher* has been used from the library called *difflib*. Each character of the keyword is compared to each word of the text. If the characters of the test word are the same as the characters of the keyword the ratio is increased. Note that the position of the characters is also checked. The ratio will range from 0 (no characters match) to 1 (all characters match). If the ratio is higher than 0.8 (80% of the characters match) the word and the ratio will be mapped in a collection. If the tested word and the keyword are of the same length a probability factor of 0.3 (30%) will be added to the ratio. This factor can be tweaked to achieve desired result.

The script has a few methods related to this topic. These methods are explained below.

When is the spellcheck called?

Not every receipt required a spellcheck in order to function. Only a handful of the receipt were in need of this procedure. The way that was found most prominent in this project was that each extraction method above were run; if one of the keywords did not have any values mapped to it the spell check was run on the keyword in question. If the total cost did not have a mapped value, the program would do a spellcheck diagnosis on the text and try to find similar words to said keyword and replace each instance where the word was found. When the spellcheck is complete, the program would execute each extraction method again.

similar

This method of the script will take two strings as input and calculate the ratio on how similar the words are. Two identical strings will not be compared since this would result in a max value of the ratio which in turn would counter the effectiveness of the following methods.

find_similar
This method has two parameters; a keyword and a text. The method will loop through each word of the text and compare them to the keyword given by the method called "similar" mentioned above. If the ratio of the words exceeds 0.8 (80%) it is considered to be a similar word. If the word is considered to be similar, it will be added to a collection containing the words and their corresponding ratio.

auto_update_most_probable
This is a method that will look at the collection of similar words of the keyword and replace each instance of the word with the highest ratio with the corresponding keyword. This method can be used for automatic processing since if a set of similar words have been found this method will initialize a sequence of method calls that will extract the word with the highest ratio and replace each instance of that word within the text.

3.3.8 Post Process

If any of the keywords related to the valuable information does not have a value mapped to it. A post process of the result is initialized where the script will calculate the mission values for the keyword using the values mapped to the other keywords. The VAT-cost can be calculated given the total cost and VAT-percentage. In a similar fashion, most missing values can be calculated. An important thing to consider is the rounding of the values. However, if we do not get the exact value we may be able to calculate it within an acceptable range.

To calculate the missing information, the relationship between total cost, VAT%, VAT, net can be considered by the following equations:

$$NET = VAT / VAT\%$$
$$Total = Net + VAT$$

Given these two equations or a rewrite of this equation each missing value can potentially be calculated given that the script was able to extract enough information.

The relationship of the values shown in the equations above can be used to perform a control analysis that the extracted values are valid. By performing the calculations given the equations above or any variation of the equations we will be able to calculate the different parts and compare it to the information that we already have extracted. This can be used to determine the validity of the extracted information.

4 Result

The amount of receipts tested in this project amounted up to 53 receipts from a 34 separate establishments. In this section the result of the project will be presented. For simplicity and easier structure, the receipts will be divided into a subset of categories. The categories are: *Lunch, Groceries, Travel, Gas and Other*. The result of each category will be presented in two ways; the first one being a table where the extracted values are displayed, and the second being a pie chart showing the success rates of the extraction. The result can be one of three outcomes; a pass, a fail, and a potentially calculated answer. The calculated answer will be marked as blue and denotes to values that can be calculated using other extracted information.

For the section of the result containing the pie chart. If the result of the extraction contains the correctly extracted values for; *total*, *vat%* and *date*, it will be considered successful. Each successful value will be marked as green. If the extracted values contain incorrect values but said values can be calculated using other extracted information, it will be marked as blue. If the extracted values are incorrect; or does not contain enough correct information to calculate the faulty values, it will be considered incorrect and will be marked as red.

Blue = Calculated value, close but not exact.
Red = Incorrect
Green = Correct

4.1 Lunch

Receipt	Total (kr)	VAT% (%)	VAT (kr)	Gross (kr)	Net (kr)	Date
1	238	12				2019-03-05
2	209	12			22.4	21/02/19
3	104	12	11.15		92.85	2018-12-12
4	66	12	6.96	65	58.04	2019-03-06
5	327	12	35.04		291.96	2019-03-07
6	109	12	11.68		97.32	2019-03-25
7	109	12	11.68		97.32	2019-04-16
8	109	12	11.68		97.33	2019-04-16
9	97.32	12	11.68		109	2019-04-05
10	60	12	6.43		53.58	2019/12/03
11	96	12	10.29		85.72	2019-03-30
12	394	6	2		371.7	07-12-2018
13	95	12	10.18		84.83	2019-04-11

14	95	12	10.18		84.83	2019-02-28
15	105	12	11.25	-	93.75	2019-04-01
16	562	12 25			360.72 126.40 158 47	2019-03-08

Table 4.1: Table of resulting values in subset Lunch after text extraction.

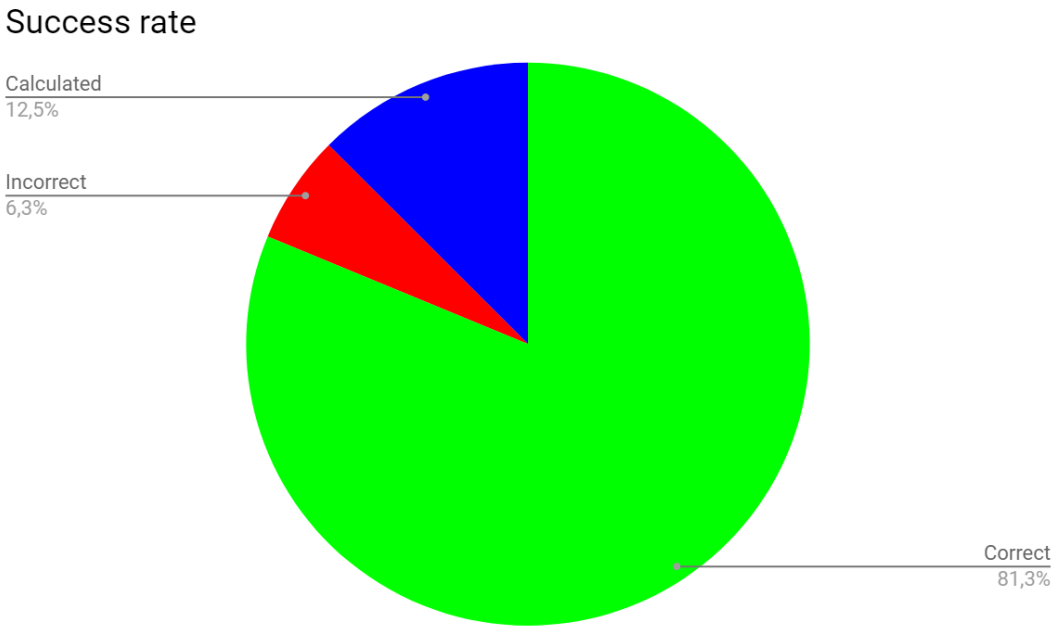

Figure 4.1: Pie chart of the result after text extraction for Mission Values from subset Lunch. Green = Correct, Red = Incorrect, Blue = Calculated

4.2 Groceries

Receipt	Total (kr)	VAT% (%)	VAT (kr)	Gross (kr)	Net (kr)	Date
1	171.6	12 25	15 18 5.99	141.65 29.95	126.47 23.96	06/10/2018
2	551.54	12 25	54.31 8.89 63.20	507.09 44.45	452.78 35.56	26/03/2019
3	551.54	12 25	63.20 - -	507.09 44.45	452.78 35.56	26/03/2019
4	400.14	12 25	35.04 292.1 14.6 49.64	327.14 73.00 58.4	-	2018-10-01
5	255.85	12 25	12.00 25.00 19.96 13.90 33.86	186.35 69.50	166.39 55.50	2019-03-01
6	92.30	12 25	9.56 92.30 10.16	89.3 3.00	79.74 2.4	2019-04-01
7	127.10	12 25	13.40	125.1	111.7	2019-03-30

Table 4.2: Table of resulting values in subset Groceries after text extraction.

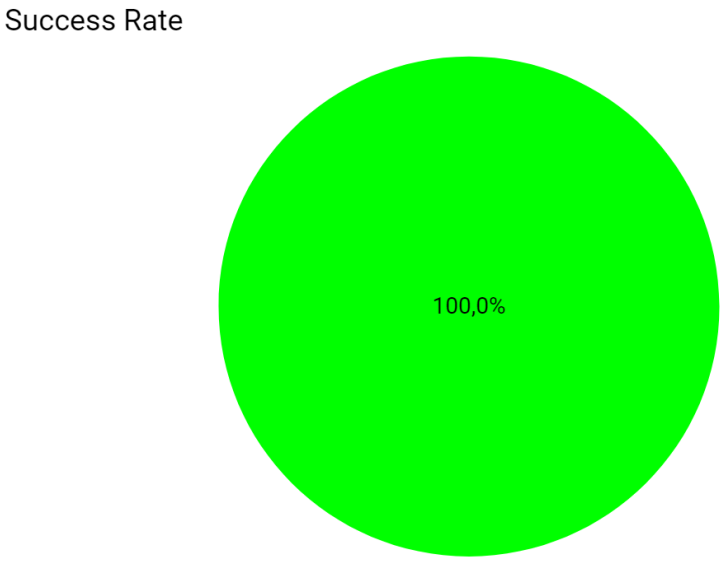

Figure 4.2: Pie chart of the result after text extraction for Mission Values from subset Groceries. Green = Correct

4.3 Gas

Receipt	Total (kr)	VAT% (%)	VAT (kr)	Gross (kr)	Net (kr)	Date
1	46	12	4.93	46.00	41.07	2019-03-21
2	835.48	25	167.1	-	668.38	2019-04-16
3	835.48	25	25.00 167. 10 668.38 835.48	-	668.38	2019-04-16
4	-	25	-	-	-	2019-02-16

Table 4.3: Table of resulting values in subset Gas after text extraction.

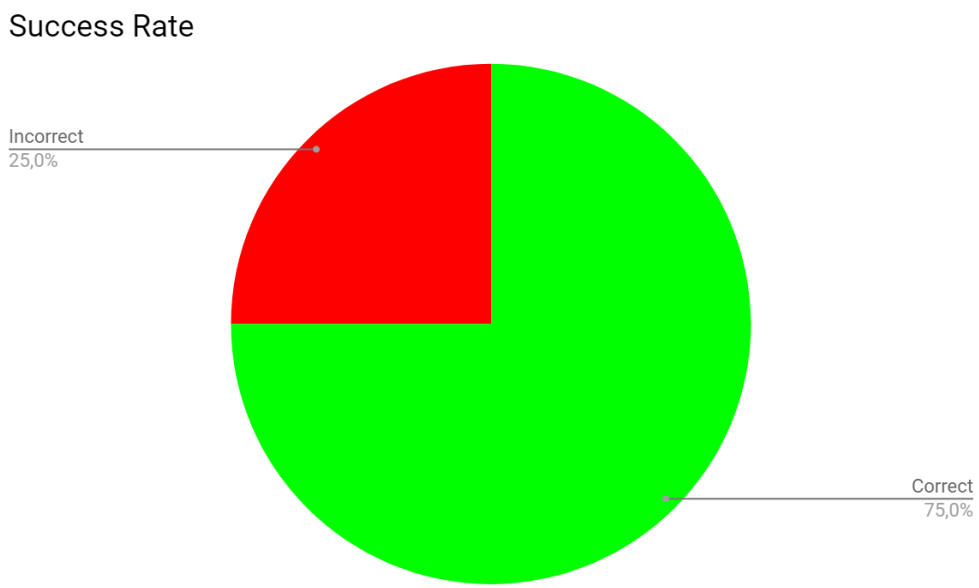

Figure 4.3: Pie chart of the result after text extraction for Mission Values from subset Gas. Green = Correct, Red = Incorrect

4.4 Travel

Receipt	Total (kr)	VAT% (%)	VAT (kr)	Gross (kr)	Net (kr)	Date
1	162	6	17		152.84	2018-12-07
2	164	6	9.29		154.72	2019-01-03
3	27					2018-11-28
4	500	6	28.3		471.7	2018-10-12
5	9215.00	6	521.6	-	8693.4	2019-03-04
6	45	6	2.55	-	42.45	2019-01-21
7	194.40	6	10.99	-	183.4	-
8	500	-				-
9	200	-				-
10	295	6	16.7		278.31	2019-04-24
11	295	6	16.7		278.31	-
12	659.00	6	37.3		621.70	2019-08-16

Table 4.4: Table of resulting values in subset Travel after text extraction.

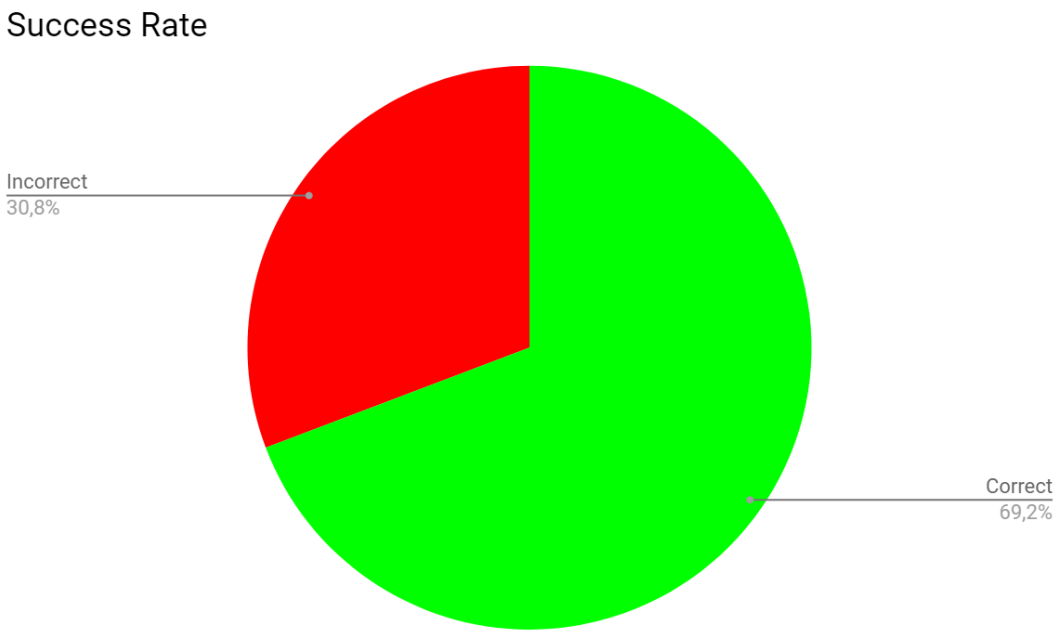

Figure 4.4: Pie chart of the result after text extraction from subset Travel. Green = Correct, Red = Incorrect

4.5 Other

Receipt	Total (kr)	VAT% (%)	VAT (kr)	Gross (kr)	Net (kr)	Date
1	275.0	25	55.00		220	2019-04-03
2	151.70	25	121.36 25.00 30.34		121.36	2019-03-18
3	779.00	25	639.20 159.90 25.00		639.21	2019-04-21
4	129.0	25	25.8		103.2	2019-04-06
5	55.0	25	11.0		44.0	
6	905.00	25 12	18.0 125.0 40.71 46.61		72.0 339.29 388.39	
7	199.0	25	39.8		159.2	
8	1010.00	25	202.0		808.0	2019-04-03
9	478.0	25	95.6		382.4	2019-03-08
10	99.00	25	99.00		79.2	19-01-11
11	105.00	25	105.00		84.0	19-04-01
12	20190302	25	72.49		16152241.6	2019-03-02 2019-04-01
13	-	-	20.0			2019-04-01

Table 4.5: Table of resulting values in subset Other after text extraction.

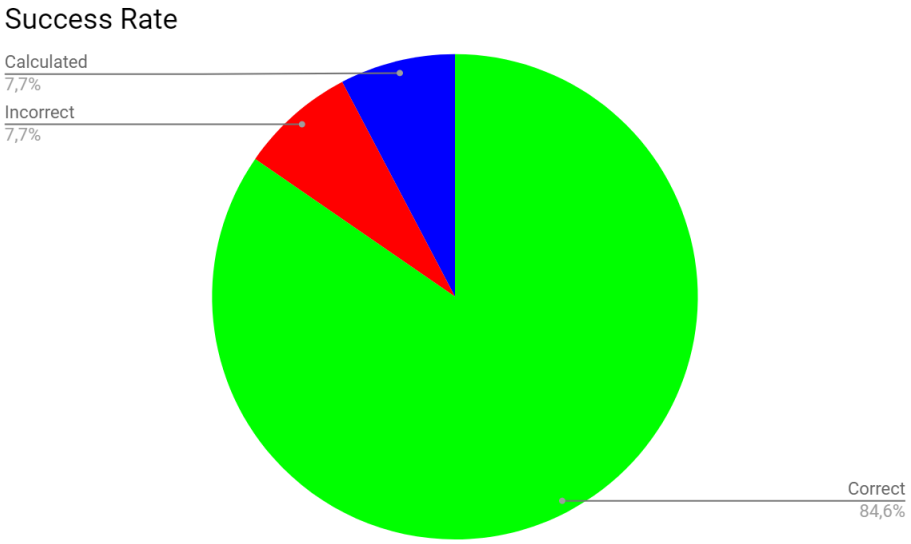

Figure 4.5: Pie chart of the result after text extraction from subset Other. Green = Correct, Red = Incorrect, Blue = Calculated

4.6 Mission Values

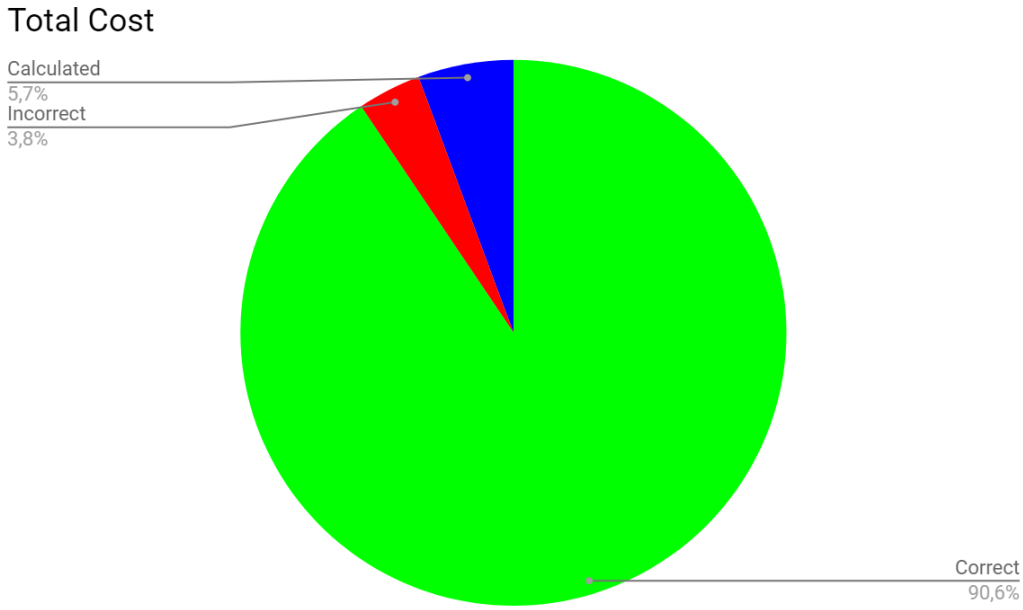

Figure 4.6: Pie chart of the result after text extraction for Total Cost from all receipts. Green = Correct, Red = Incorrect, Blue = Calculated

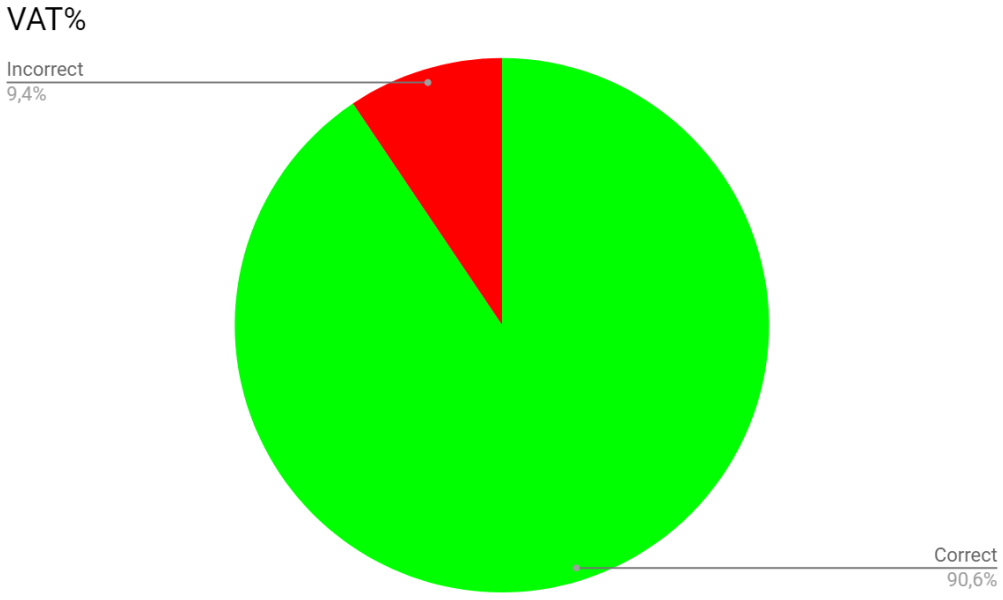

Figure 4.7: Pie chart of the result after text extraction for Total Cost from all receipts. Green = Correct, Red = Incorrect, Blue = Calculated

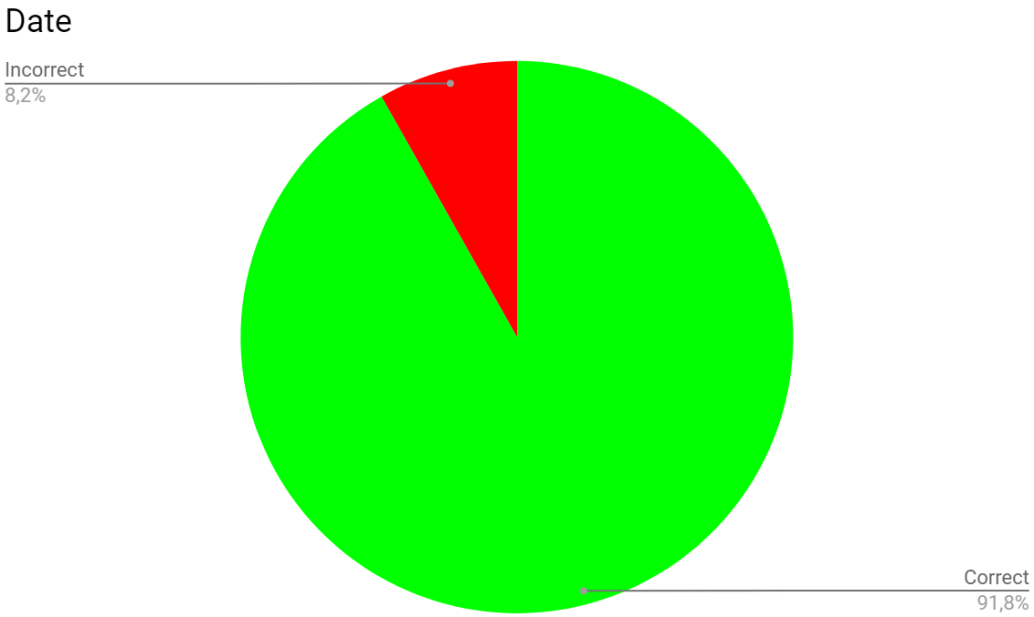

Figure 4.8: Pie chart of the result after text extraction for Total Cost from all receipts. Green = Correct, Red = Incorrect, Blue = Calculated

4.7 Total

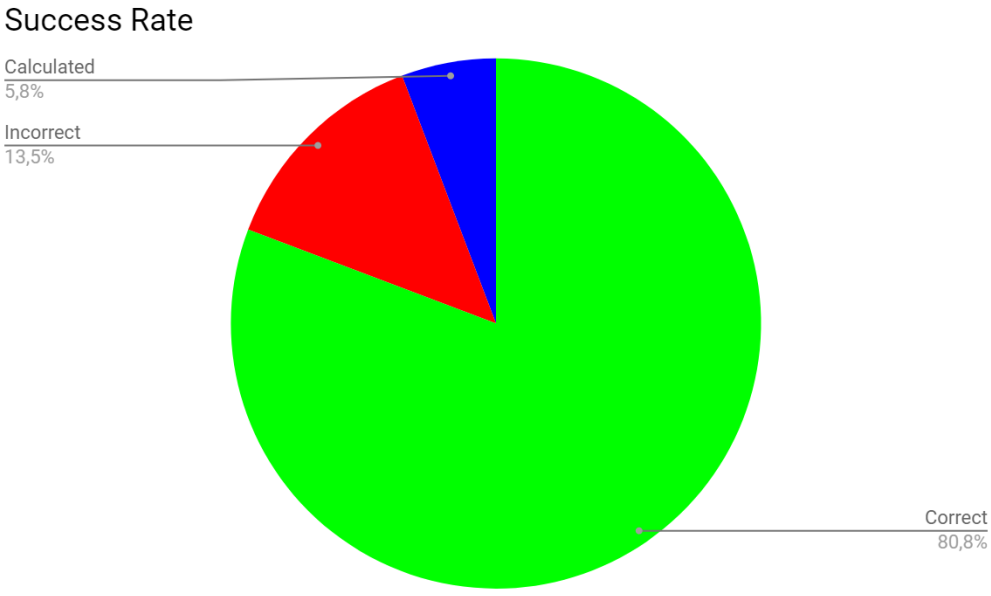

Figure 4.9: Pie chart of the total result after text extraction for Mission Values. Green = Correct, Red = Incorrect

4.8 Pre-processing

In this section the result of pre-processed images will be presented. Each of the receipts that were tested are receipts that the program initially failed to extract the valuable data from. The result will be presented in a set of tables where the result of each pre-processing step is presented along with the expected result for easy comparison.

4.8.1 Receipt 1

Method	Total (kr)	VAT% (%)	VAT (kr)	Gross (kr)	NET (kr)	Date
Expected Result	835.48	25	167.10	835.48	668.38	2019-04-16
None	835.48				25.00 167.10 668.38 834.48	2019-04-16
0.5x Resize	835.48	25	25.0 167.1	835.48	668.38	2019-04-16
1.5x Resize	835.48	25	835.48		668.39 3341.92	2019-04-16
2x Resize	835.48	25	835.48	835.48	668.38 3341.92	2019-04-16
Greyscale	835.48	25	835.48		668.39 3341.92	2019-04-16

Table 4.6: Results of extracted information from a receipt after pre-processing has been applied.

4.8.2 Receipt 2

Method	Total (kr)	VAT% (%)	VAT (kr)	Gross (kr)	NET (kr)	Date
Expected	109	12	11.68		97.32	2019-04-05
None		12				2019-04-05
0.5x Resized	97.34	12	11.68		86.92	2019-04-05
1.5x Resize	109.00	12	11.68		97.32	2019-04-05
2x Resize	109	12	11.68		97.32	2019-04-05
Greyscale		12				2019-04-05

Table 4.777777: Results of extracted information from a receipt after pre-processing has been applied.

4.6.4 Receipt 3

Method	Total (kr)	VAT% (%)	VAT (kr)	Gross (kr)	NET (kr)	Date
Expected	400.14	12 25	35.04 14.60	327.14 73.00	292.10 58.40	2018-10-01 01/10/2018
None	400.14	12	35.04 14.60 292.10 58.40 327.15 73.00 400.14		357.27	2018-10-01 01/10/2018
0.5x Resized	400.14	12 25	49.64	327.14 -	73.00	2018-10-01 01/10/2018
1.5x Resized	400.14	12	49.64	327.14 73.00 35.04 292 10 14.60 58.40 400.14	357.27	2018-10-01 01/10/2018
2x Resized	400.14	12	12.00	327.14 73.00	292.10 14.60 58.40 35.04 400.14	2018-10-01 01/10/2018
Greyscale	400	12	49.64	327.14	73.00	2018-10-01 01/10/2018

Table 4.8: Results of extracted information from a receipt after pre-processing has been applied.

4.6.5 Receipt 4

Method	Total (kr)	VAT% (%)	VAT (kr)	Gross (kr)	NET (kr)	Date
Expected	500.00	6	28.30		471.70	2018-10-12
None	28.3	6	1.7 500.0		47.0	2018-10-12
0.5x Resized	500.00	6	28.3		471.7	2018-10-12
1.5x Resized	500.00	6	1.7 500		47.0	2018-10-12
2x Resized	500.00	6	500.0		471.7	2018-10-12
Greyscale	500.00	6	29.3		471.7	2018-10-12

Table 4.9: Results of extracted information from a receipt after pre-processing has been applied.

4.8.3 Receipt 5

Method	Total (kr)	VAT% (%)	VAT (kr)	Gross (kr)	NET (kr)	Date
Expected	95.00	12	10.18			2019-04-11
None		12				2019-04-11
0.5x Resized		12				2019-04-11
1.5x Resized	95	12	10.18		84.83	2019-04-11
2x Resized		12				2019-04-11
Greyscale		12				2019-04-11

Table 4.10: Results of extracted information from a receipt after pre-processing has been applied.

5 Analysis

5.1 General Results

The values corresponding to the values; *total, vat% and date* must be correct in order for the extraction to be considered successful. Given this information each of the other values can be calculated. Three colours of the values can be seen if the result is studied; green, red and blue. If a value is represented by a green colour it is a correct extraction. If the value is red the extraction is incorrect. And lastly, if the value is represented with a blue colour it is a calculated value. If a value is calculated the rounding of the number must be considered. By rounding values, the calculated value may not be the exact same value as the correct value. But in the tables above the calculated value will be within 0.01 SEK of the correct value. In some cases, the values can be calculated using the other extracted information; even though the program initially failed to extract the correct information. With that in mind, some of the values in the table are marked with a red colour, representing an incorrect value. If the value can be calculated, they will be marked with a blue colour in the pie-charts representing the results.

The result seen in figure 4.9 can be interpreted as a success rate of 86.6%. This success rate can be achieved if the calculated values marked as blue are considered to be correct.

The reason for the incorrect values can be manifold. In at least four of the cases above the program failed to extract the correct date. In this case the reason for the failed extraction was that the date on the receipt was represented with an unimplemented structure. The date on these receipts was of the following structure:

03.apr 19

This structure has yet to be implemented in the program. In the cases where the above-mentioned structure was encountered, the value for total cost and VAT% was successfully extracted. If this structure was to be implemented the total success rate would be increased to 94.3%. Procedures that may increase the accuracy of the program will be discussed further in the section *Future Work* of the report.

In the cases where the program failed to extract the data, the most common reason was due to unsupported text structure. The text structure has close resemblance to the supported structure but did not have the exact same structure. This might have been solved by retaking the photo or performing pre-processing of the images.

The last objective to cover was to evaluate the finished solution, this can be done using the information in the result section of this report. Each table and figure contain information on how accurate the extraction was; different values are represented with three different colours. The values marked with green colour are correct information, values marked with blue colour are values that can be calculated using existing values and the values marked with red colour represent incorrect information.

5.2 Pre-processing Results

As seen in the tables at the end of the result section where a few pre-processing methods were tested. The pre-processing methods were applied on receipts that initially failed to produce desirable results. These receipts were chosen to see if pre-processing of the images can be used to increase the success rate of the text extraction. As seen in the tables 4.9 through 4.13; pre-processing can be applied to increase the success rate of the text extraction. Different pre-processing methods worked best for different receipts. For a

receipt to produce sufficient result they have to produce values such as; total cost, date and VAT%. Or information that makes it possible to calculate those values. Since different pre-processes methods produced the desired result it might be hard to conclude what pre-processing method to use. A combination of all methods will result in an increased response time since for each pre-processing method will result in a new image that has to be send to the OCR engine.

5.3 Mission Values Results

The result of the extraction of the Mission Values was presented in section 4.6 of this report. As seen by the figure 4.6 describing the extraction of the *Total Cost*; the success rate of this value was the highest success rate amongst all the mission values. Since the calculated values marked as blue are considered to be correct, the success rate of total cost amounted to 96,3%.

The result of the extraction related to *VAT%* amounted to 90,6%, with no calculated values. In the occasions where the script failed to extract the correct values related to VAT% the script also failed to extract enough information to calculate VAT%.

The result of the extraction related to *Date* amounted to 91,8%, which is slightly higher than the result of the VAT% but significantly lower than the result of the total cost. By studying the tables in the result section of this report we can see that in the occasions where the script failed to extract a mission value the script also failed to extract another mission value from the same receipt. The reason for this can be manifold; the image used to extract the text from may not meet the requirements for the OCR-engine to properly extract the correct values, which in turn will drastically affect the result of the script. Some methods can be implemented in order to increase the success rate of the text extraction. As seen in the result of the pre-processing section of this report; pre-processing may be used in order to increase the conditions of the OCR-engine to extract the correct information which in turn will yield increased conditions for the script to extract the valuable information.

Pre-processing may not work in every case. As mentioned in the General Result section, a new type of text structure was encountered that was not supported by the script. This proves that in some cases, increased quality of the image may not always translate to an increased success rate of the text extraction. By implementing this text structure, the result can effectively be increased.

Even though the success rate of the extracted values related to *Date* and *VAT%* was similar in terms of percentage, the failures did not always occur on the same receipts.

Some of the occasions where the script failed to extract some data could be averted if more time and resources was spent on this project. If I had more time to spend on this project, but only enough time to spend on one of the mission values I would choose to invest the time in the mission value *Date*. The reason being is that some receipts failed due to an unsupported text structure, mentioned in section 5.1 General Results. An increased success rate could be achieved by implementing this structure. Another argument for investing more time in the extraction related to date is that this is the only mission value that cannot be calculated using the rest of the so-called *valuable information* as well as the success rate of this mission value was amongst the lowest. The same argument can be used in favour of the other mission values; by increasing the success rate of one mission value such as VAT% another mission value such as Total cost can be calculated. However, since all three mission values must be correct to achieve a successful extraction I would invest more time in the extraction of the values related to date.

6 Discussion and Conclusion

The expected result of this project was not intended to be a fully functional application but rather an initial program in order to determine whether or not it is possible to create a sufficiently accurate program in a restricted time frame. An initial concern was that a large data set of receipts was required in order to create a seemingly dynamic program. Something that was not available. The receipts used in this project was gathered by the employees of the company and a few personal receipts of mine. Even though the amount of receipts used was restricted to only 53; they originated from 34 separate establishments. Given the widespread areas or origin and the similarities of the text structures one can determine that if a sufficient extraction can be achieved on this set of receipts other untested receipts have a high probability to also work. In the cases where the program failed to extract all correct valuable information some was still extracted. Even though the program failed to extract information in some cases; the extracted data could still be used in order to automate parts of the process.

There are a few steps of this project that could have been done differently in order to achieve higher success rates. One of the most prominent factors to be considered is the set of receipts used. In order to see the true potential of the program it should be tested on more receipts. By increasing the number of receipts and the origin of the receipts the program could be perfected to handle more structures. An increase of tested receipts could even yield a more precise result of the success rate for extracting data. Another action that may yield better result is to take picture of receipts with higher quality; some of the receipts tested in this project had its text almost completely faded out which may yield worse result of the text extraction. Even though the quality of the OCR proved to be good even more time should be invested in pre-processing of the images. As seen the result section of this project pre-processing can be used to increase the likelihood of correctly extracted data. Given that different pre-processing methods yielded better result depending on the structure of the receipt, there are not enough information to conclude which pre-processing method that should be used. A compromise in response time may be necessary in order to yield the best result since more methods may be needed. In this project the pre-processing methods tested are grey scaling and resizing of the image.

Given the success rates achieved in this project and the wide variety of receipts used one can conclude that it is possible to develop a program that can extract valuable information given limited resources. Even though the result of this project is preliminary, and more receipts must be tested to perfect the program, it is proven be feasible to do.

7 Future Work

Given the limited timeframe and resources of this project there are a few things that can be improved. This project was developed using a limited amount of receipts. To improve the program even further the first thing to do is to test the program on more receipts to see if the structures found in this project can also be found in other types of receipts.

Even though the OCR used in this project produced a good result; further investigation in pre-processing may be necessary. As seen in the result section; implementing pre-processing on images that have initially failed to yield acceptable data extraction resulted in a successful data extraction. The receipts tested in the pre-processing section was put through several pre-processing methods. Different methods yielded better result for different types receipt. To increase the effectiveness program, further investigation of how the methods correlate to the receipt structure may be needed.

The pre-processing step was only run on receipts that initially failed to produce acceptable result. Each of these methods were carried out manually. To automate the process even further the receipts that succeeded to produce acceptable results should also be run though the pre-processing steps to examine if the results are any different. If the results stay the same a mandatory pre-processing step can be implemented to increase the success rate of the data extraction.

References

[1] [TechTerms], [*OCR Definition], [https://techterms.com/definition/ocr*], Visited: [2019-03-29]

[2] [Raamesh Gowri Raghavan, Prashant Manoharrao Kakde, and Sukant Khurana], [Medium], [url: https://medium.com/swlh/applications-of-ocr-you-havent-thought-of-69a6a559874b], Visited: [2019-03-29]

[3] [AI & Machine Learning Products, Cloud Vision], [Google Cloud], [https://cloud.google.com/vision/], Visited: [2019-04-05]

[4] "Image Preprocessing for Improving OCR Accuracy", Wojciech Bieniecki, Szymon Grabowski, Wojciech Rozenberg", Publisher: [IEEE], Published in: [2007 International Conference on Perspective Technologies and Methods in MEMS Design], Visited: [2019-05-23], url: [https://ieeexplore-ieee-org.proxy.lnu.se/document/4283429]

[5] "Image processing based degraded camera captured document enhancement for improved OCR accuracy", "Pooja Sharma, Shanu Sharma", Publisher: [IEEE], Published in: [2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)", Visited: [2019-05-23], url: [https://ieeexplore-ieee-org.proxy.lnu.se/document/7508160/references#references]

[6] [Yasser Almodhi], [Classifying Receipt and Invoices in Visma Mobile Scanner], [url:http://lnu.diva-portal.org/smash/record.jsf?pid=diva2%3A901992&dswid=-5151], Visited: [2019-04-10]

[7] "Pillow Documentation", "Pillow", Visited: 2019-05-15, url: https://pillow.readthedocs.io/en/stable/

[8] "Improve OCR Accuracy With Advanced Image Processing", "DocParser", Visited: [2019-05-13], url: https://docparser.com/blog/improve-ocr-accuracy/

[9] "Receipt", "Wikipedia", Visited: 2019-05-13, url: https://en.wikipedia.org/wiki/Receipt

[10] "Improve Accuracy of OCR using Image Preprocessing", Author: Brijesh Gupta, "Cashify Engineering", Visited: 2019-05-13, url: https://medium.com/cashify-engineering/improve-accuracy-of-ocr-using-image-preprocessing-8df29ec3a033

[11] "A Simple Introduction to DPI", "SnapShop", Visited: 2019-05-13, url: https://snapshop.cam/dpi/

[12] "Dots Per Inch", "Wikipedia", Visited: 2019-05-13, url: https://en.wikipedia.org/wiki/Dots_per_inch

[13] "Grayscale", "Techopedia", Visited: 2019-05-13, url: https://www.techopedia.com/definition/7468/grayscale

[14] Docs Overview for Projects, Google Cloud, visited: 2019-05-11, url: https://cloud.google.com/docs/overview/#projects

[15] “Lossy Definition”, “TechTerms”, Visited: 2019-05-15, url: https://techterms.com/definition/lossy