Abstract

The purpose of this final year project is to create an Android mobile application that can automatically extract relevant information from pictures of receipts. Users can also load their own images from memory as long as they are pictures of receipts. For this particular mobile application only the english language will be supported.

Problem Statement

Whenever you make a purchase, you will likely have received a physical receipt. However, what do you normally do with the receipt? Throw it away? Mark down important information? Take a photo of it? It is rare for a customer to take time and enter the information into a software. Even if it is necessary, for business or tax purposes, everyone agrees that it is a hassle to have to input information by hand. Of course, there are existing solutions to this problem, for example Smart Receipts[1] and Receipts by Wave[2]. However, most applications that tackle this problem come with various limitations like:

1. You need to subscribe to the service and pay a monthly fee
2. There are a lot of advertisements
3. Receipts cannot be recognized unless all four corners are visible and the image is vertical
4. The processing is done in a server, ie there are additional bandwidth costs
5. The applications only display the information in tables

Objective

In order to address the above issues, a free Android mobile application will be released to the Google Play Store. There will be no subscriptions or advertisements, and the processing will be done in the device in real time. The application will attempt to automatically detect the receipt and will only ask the user to mark the corners if the process fails. This will allow the user to upload partially covered receipts and still make use of the mobile application. After extracting relevant information (Location, amount spent, date and time), they will be shown to the user, where they can make further changes and/or add additional information. All of this will then be show in interactive graphs for the user. They can also set a custom spending target to be automatically reminded whenever they are nearly over their budgets.

Benefits

Although existing mobile applications may not be too reliable, some may ask why do we need such an application in the first place? By automating the process of information input, not only will time be saved, but it will also reduce the percentage of human error. Not only that, as users can modify the automatically detected information, the system will be able to improve over time. Digitising the information will also ease the process of data analysis. Particularly in company environments,
spendings affect multiple aspects of operations. Digitized content is also more safe as backups can be made.

**Basic Background**

There are 2 main parts of this project: Optical Character Recognition and the actual Android mobile Application. Optical Character Recognition is the process of identifying printed characters/symbols from digital sources such as images, and converting them to software-friendly representations. It can be further subcategorized into 3 parts: document layout analysis, character recognition and information extraction. This process is simply identifying text and relevant symbols before characterizing them into alphabet, number or symbol. After characterization groups of letters will form words, and words will form sentences.

After this information is collected, it will be passed to the Android application. Its main use is to display the information to the user as well as take user input, whether that be gestures or image files.

**Theoretical Background**

Optical Character Recognition is the process of extracting characters and symbols from an image and converting them to a machine-readable format. Although this process applies to both printed and handwritten text, this project will only focus on printed characters, specifically those on a receipt. Extracting only the text from an image can be difficult because computers only “see” 1s and 0s for an input image. A picture of a receipt simplifies this process a little as we know that the target information will be contained inside a rectangular area. Furthermore, receipts follow similar patterns, with the company name at the top, items purchased in the middle, followed by the total and the payment method at the bottom. Of course, every company will have its own format for the receipts, which is why document layout analysis is vital. This is the process of identifying useful regions of information while discarding non essential data at the same time. Following this step is to segment regions into individual characters/symbols for categorization. Due to the huge number of fonts and size of text, it is impossible to predict how the information extracted will look like. Machine learning enables us to “train” a program to differentiate and identify relevant characters. The more data we provide the program, the more accurate it becomes. Unfortunately, this process will never be 100% accurate and that’s why it will only serve as a tool to simplify the conversion process, not replace it. Even if we were able to correctly identify every picture, we still need to make use of the information. Again, as we are only working with receipts, there will be common patterns, for example, numbers following the dollar sign will most likely be the price, rather than the telephone.

**Literature Review**

Optical Character Recognition is not a new topic but it is certainly one that continues to be a topic of interest for researchers. Most research papers will focus on one or two machine learning
algorithms and either compare their strengths and weaknesses or propose a novel alternative solution that modifies an existing algorithm.[3,4] However, there are few papers which make use of two or more algorithms simultaneously. This project will attempt to make use of 3 algorithms in order to improve the accuracy of optical character recognition

Scope

Only an Android version of the mobile application will be developed. The software that does the optical character recognition will be embedded into the application.

Prerequisites

Hardware: Android devices
Software: OpenCV,
Techniques: Supervised machine learning algorithm (Support Vector Machine, K-Nearest Neighbour)

Deliverables

- Optical Character Recognition Algorithm
- Android application that makes use of above algorithm

Approach and Methodology

The Optical Character Recognition Algorithm will first be written in Python using the OpenCV library version 2. Although OpenCV 3 is the latest version at the time of writing, it will not be used as there is an ongoing bug regarding saving and loading models of Support Vector Machine. Also, most OpenCV projects and tutorials are written for C++ and Python so it will be easier to find relevant resources and help if necessary. It will be converted to Java when used for the Android mobile application.

In optical character recognition, the quality of input is vital to the accuracy of the software. As the receipts will be taken using a mobile camera or uploaded using the gallery, there will be a lot of “noise”. There will be different lightings, creases, smudges and even blurred areas. In order to filter out all these useless obstacles, binarization will be performed. This is the process of converting every pixel of the image to “black” or “white”, depending on the intensity of the pixel. After this process is completed, we will need to extract the receipt from the image. If a contour with 4 corners is found, this is assumed to be the receipt and only that subsection of the image will be extracted for further processing. As users may take the image with part of the corner obstructed, this step may fail and so we will need the user to manually mark the corners.
After the receipt is isolated, the image will be inverted so that the background is black and the text/symbol is white. A process called dilation will then be applied. This makes all white areas expand outwards, creating “blobs” of text. Using the “blobs” as reference, sub-images will be extracted. From each sub image, individual characters will be isolated and extracted. Each character will then need to be categorized using a machine learning algorithm. There are 3 main types of machine learning and this project will make use of supervised learning. This means for every input, there is a specific output we want, in this case: alphabets (lower- and upper-case), numbers (0-9) and symbols (at symbol-@, dollar sign-$ etc). The last part, information extraction, will be the most difficult. This is because after we recognize individual characters, we need to meaningfully piece them back together like a jigsaw puzzle. Do numbers represent the price, telephone, address, email, company name, date or quantity? In order to address this, we will first sort the contours from left-to-right and top-to-bottom. Then we will analyze the neighboring contours to predict common patterns. For example, numbers in groups of 8 will likely be telephone numbers while strings with the “@” symbol will likely be part of an email address. This information will then be passed to the Android application for display.

**Feasibility Assessment**

Optical character recognition is still an ongoing research topic. Particularly for information such as receipts, it is difficult to account for the numerous font text and sizes. To train the “algorithm”, it is necessary to supply it with numerous training examples. However, there is no default database with which to collect relevant information. Hence, it is necessary to generate random receipt data.

Nonetheless, with the number of mobile applications already released specifically for receipt recognition, it will definitely be possible to create such an application, though the accuracy will still need to be analyzed.

**Schedule**

<table>
<thead>
<tr>
<th>Period of Time (2016-17)</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>September</td>
<td>Research Optical Character Recognition</td>
</tr>
<tr>
<td>October</td>
<td>Implement first part of Optical Character Recognition in Python using OpenCV library. (Text extraction and isolation)</td>
</tr>
<tr>
<td>October</td>
<td>Improve above to account for noise factors</td>
</tr>
</tbody>
</table>
October-November | Implement second part of Optical Character Recognition in Python using either OpenCV or Tensorflow. (Character recognition). A KNN and SVM model will be trained in OpenCV and if their performance is not satisfactory then alternative methods will be pursued.

December | Migrate code from Python to Java for use in Android

December-February | Create Android application to merge with optical character recognition code.

February-March | Test and improve android application

**Mini Conclusion**

This project will create an Android mobile application that can extract relevant information from the picture of a receipt.

**Appendices and References**