Building an Expense-based Transaction History using Receipts and OCR

Kevin Burgon

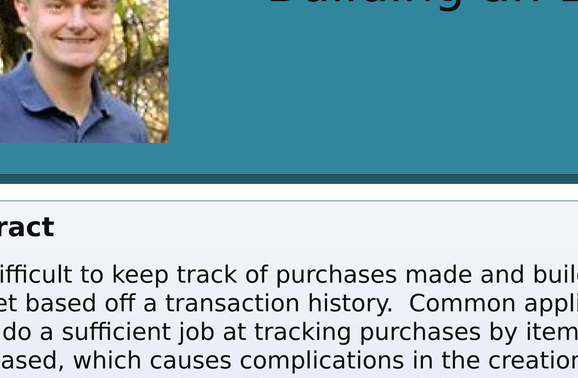
CS 5600 Fall 2016

Abstract

It is difficult to keep track of purchases made and build a budget based off a transaction history. Common applications don't do a sufficient job at tracking purchases by items purchased, which causes complications in the creation of a well-categorized budget. The purpose of this project is to use Optical Character Recognition to keep a categorizable transaction history based off of the text pulled from receipt images.

Problem Description

One of the more difficult parts of keeping a budget is tracking expenses and splitting them into different categories. This task is made even more difficult when purchases are made in one transaction that include individual items that should be split into different categories in one's budget. Applications exist that will track and categorize transactions made in bank accounts, but these applications don't take into account those transactions that include items that belong to different categories.



Purchased items that may belong to different categories in a budget

Proposed Solution

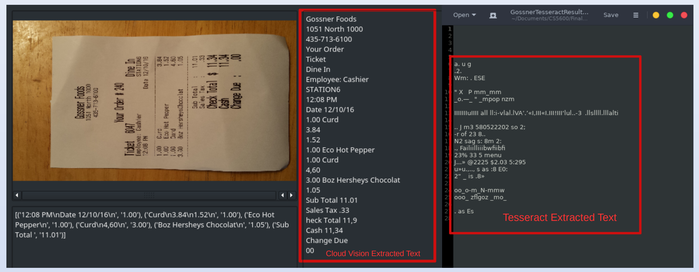
Build an application which can read text from receipt images and track expenses per item purchased instead of tracking by total spent at purchase.

Method

An experimental application was created using python that reads receipt images, collects all the text read from the image, and generates a list of the items purchased and their prices. An OCR engine was used to extract text from the receipt images, and regular expressions were used to find purchased items and their prices.

Results

Choosing an OCR Library: The original plan for the project was to use the open source OCR engine called Tesseract to read receipt images. It was quickly discovered that Tesseract is not extremely accurate and that a fair amount of image manipulation would be needed to get even semi-accurate results.

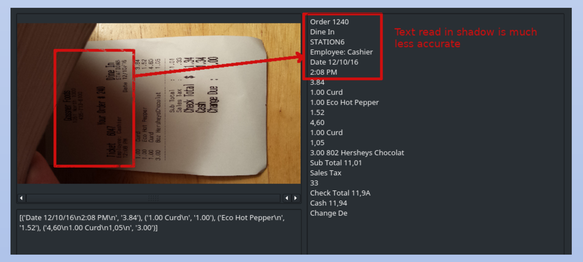


Extracting text from a receipt using Google Cloud Vision vs. using Tesseract

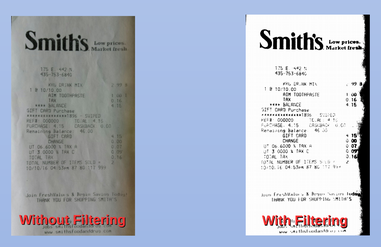
Google Cloud Vision API: While Google Cloud Vision isn't free or open source, it has some advantages over OCR engines like Tesseract that made it a much better option to use in this project. Text can be pulled with a high degree of accuracy from an image under good conditions.

Challenges

Reading Text Under Different Lighting Conditions: While Google Cloud Vision is accurate under good lighting conditions, shadows or overall darker lighting cause for some difficulty in singling out text in an image.



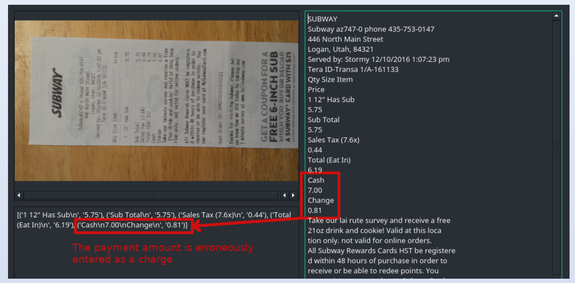
Extraction run on an image with a shadow covering part of it

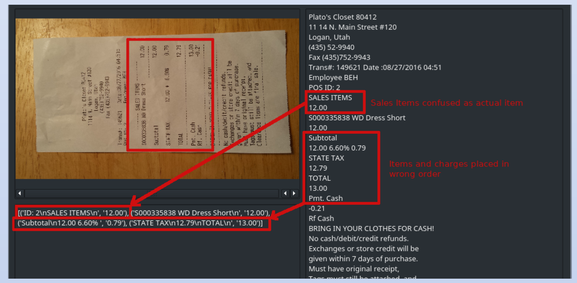


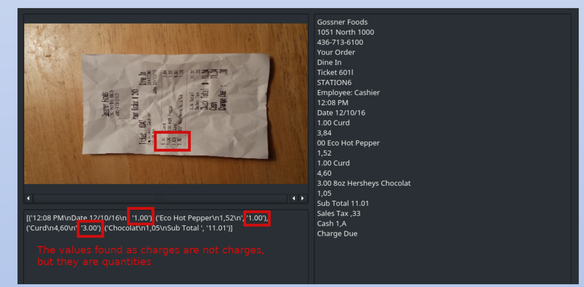
Threshold filtering does not improve image readability

Results (Continued)

Item Recognition: A variety of receipts from eight different businesses were tested, and challenges suggested that generally pulling individual charge information from a set of all receipt text is not reliable and never provides completely correct results.







Conclusion

By using the Google Cloud Vision API and regular expressions, an application was made that could read some purchases and prices from receipt images. Future improvements will include algorithms to learn items purchased and their prices from text locations and user input.

References

[1] Google Cloud Vision API, https://cloud.google.com/vision/, Last Accessed 12/03/2016

[2] Tesseract, https://github.com/tesseract-ocr/tesseract/blob/master/README.md, Last Accessed 12/10/2016