# Automated Receipt Image Identification, Cropping, and Parsing

Alex Yue
Princeton University
acyue@princeton.edu

## Abstract

*The identification and extraction of unstructured data has always been one of the most difficult challenges of computer vision. This form of data exists all around us, from flashy advertisements to basic nutritional information on food products, and provides a large amount of information that exists nowhere else. Parsing this sort of data is very challenging; however, recent advancements in computer vision technology help make this feasible.*

*In this paper, I introduce a novel data pipeline for the identification, cropping, and extraction of unstructured data within receipt images, a ubiquitous, commonplace item that many consumers receive, but one that is difficult to be transformed into raw data. Receipts contain a dense amount of data that is useful for future analysis, but there exists no widely available solution for transforming receipts into structured data. Existing solutions do exist; however, they are either very costly or inaccurate.*

*The pipeline that is described in this paper outperforms existing solutions by a large margin, and offers the ability to automatically pull out semantic data such as transaction date and price from an image of a receipt. It achieves this success by using Line Segment Detection (LSD), Holistically-Nested Edge Detection (HED), and Hough Transform to crop the image. Then, Optical Character Recognition (OCR) is applied to detect chunks of text and process the cropped image. Finally, natural language processing and statistical techniques are used to extract useful sets of information from the recognition result.*

## 1. Introduction

The task of unstructured data processing has received growing interest over the past few years as computer vision and machine learning algorithms have significantly improved. In particular, this paper is interested in the processing of unstructured receipt image data and converting it into a simple-to-use, analyzable structured data format. This problem is extremely interesting due to the fact that it is *open-ended* and the search space is unending: a receipt image can be provided in an unstructured format that varies widely across geographies and industries, and data such as date and price must be extracted from all sorts of receipt types. An airline electronic ticketing receipt is much different from a grocery store receipt, with both containing similar sets of data in drastically different locations.

This paper aims to help resolve some of the fundamental difficulties and inconsistencies associated with parsing this sort of unstructured data. We use a combination of receipt image datasets provided by ExpressExpense [2] and custom receipt data collected over the past few months by a small group of people to train and evaluate the effectiveness of this system. Our aim is to be able to process and parse receipt data from most standard use cases, which involves steps such as correcting the input image orientation, cropping the receipt to remove the background, running optical character recognition (OCR) to pull text from the image, and using heuristics to determine relevant data from the OCR result. A good, satisfactory system should be able to handle the above characteristics and further edge cases by pulling the transaction date and price from image inputs. The paper will demonstrate how this result was achieved through which processes and provide both a quantitative and qualitative evaluation with respect to the success of this system.

## 2. Related Work

The challenge of automated receipt image data extraction has mostly been taken up with respect to two regards: academia and industry. Researchers and academics have mostly focused on developing techniques to improve the recognition and extraction of text from unstructured data whereas industry has focused on creating commercial systems to reduce manual labor costs associated with inputting receipt image data for analysis or reporting. However, neither produce a optimal system due to degradations in either accuracy or cost. Existing commercial solutions include implementations from companies such as Expensify, Metabrite, and Receipts by Wave. These implementations are all slow, costly, and still rely on manual labor to process edge cases.

## 2.1. Academic Research

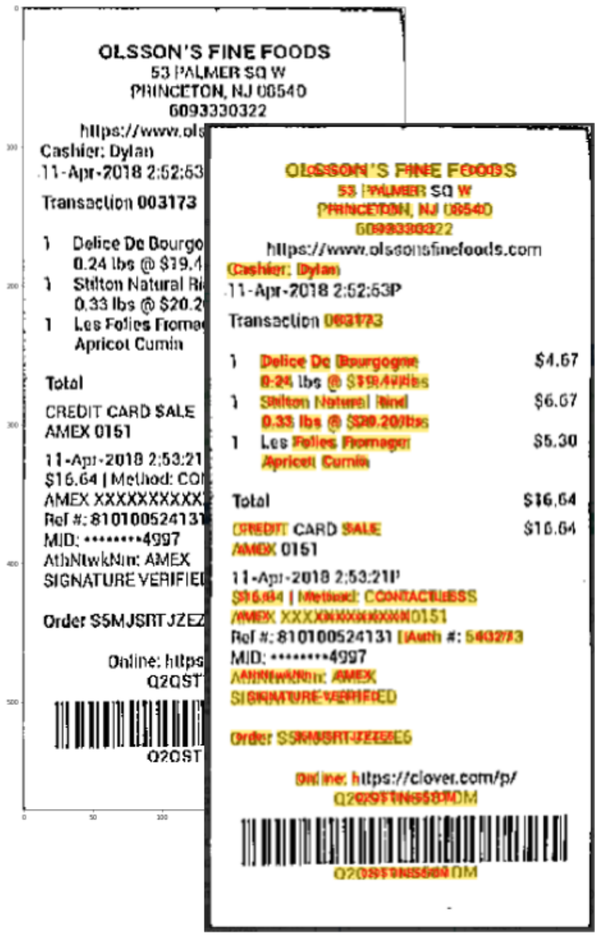
A large amount of research has been focused around optical character recognition and text extraction from unstructured data. Less has focused specifically on the subtask of data extraction from receipts. One commonly used extraction mechanism for text detection is to use a Convolutional Neural Network (CNN) [9] [7] [4]. This class of OCR utilizes a Region Proposal Network (RPN) to propose regions of interest where text may exist as well as a CNN to determine the likelihood of text appearing at that location. These systems all provide end-to-end for text identification—from localizing and recognizing text in images to retrieving data within such text. In the specific case of ”Multi-Oriented Text Detection with Fully Convolutional Networks” by Zhang et al., their proposed system consistently achieved state-of-the-art performance on numerous text detection benchmarks [9].

Some research work has also been done on direct receipt text sentiment extraction as well. There exists research by Gjoreski et al. regarding ”Optical Character Recognition Applied on Receipts Printed in Macedonian Language” [3]. In this work, the team proposed utilizing a $k$-nearest neighbors classifier to classify individual chunks of images of text that was extracted from receipt images. This approach achieved a 87% accuracy, which is decently performant for unsupervised data extraction.

## 2.2. Commercial Implementation

There are numerous companies that implement basic receipt parsing methods in order to collect structured data from user-uploaded receipts. Most notably, the leader in the field at receipt data extraction is Expensify, an integrated, enterprise expense report solution. Expensify markets their receipt scanning solution as SmartScan, which provides a service that will ”lift the Merchant, Date and Amount and create an Expense on your behalf” for a given image [1]. This system is not perfect, and relies heavily on human intervention for their numerous failure cases. More specifically, Expensify outsources some amount of their data entry collection process out to freelancers on Amazon Mechanical Turk.

Other solutions freely available on the market are offered by companies such as Wave, Shoeboxed, and Metabrite. These systems all provide rudimentary solutions that are not able to process many different types of user-uploaded receipts. In the case of Wave, it is only able to extract data from simple, uncrumpled receipts on a solid background. Other solutions such as Metabrite are expensive, slow, and unreliable.



OLSSON'S FINE FOODS
53 PALMER SQ W
PRINCETON, NJ 08540
6093330322
https://www.olssonsfinefoods.com
Cashier: Dylan
11-Apr-2018 2:52:53P
Transaction 003173

| | | |
|---|---|---|
| 1 | Delice De Bourgogne 0.24 lbs @ $19.49/lbs | $4.67 |
| 1 | Stilton Natural Rind 0.33 lbs @ $20.20/lbs | $6.67 |
| 1 | Les Folies Fromager Apricot Cumin | $5.30 |
| | Total | $16.64 |
| | CREDIT CARD SALE | $16.64 |

AMEX 0151
11-Apr-2018 2:53:21P
$16.64 | Method: CONTACTLESS
AMEX XXXXXXXXXXX0151
Ref #: 810100524131 | Auth #: 540273
MID: ********4997
AthNtwkNm: AMEX
SIGNATURE VERIFIED
Order S5MJSRTJZEZE6
Online: https://clover.com/p/
Q2QSTNS58TDM
Q2QSTNS58TDM

Figure 1. Demonstration of automated receipt cropping and sample OCR recognition.

# 3. Methodology

This research project is comprised of numerous methods and systems that work together in order to automate the extraction of data from receipt images. The primary steps involved in this systems project include: identification of receipt foreground and cropping, computation of affine transformation to unwarp the image, extraction of textual information using OCR, and, finally, the parsing of data to generate structured, semantic meaning.

The approach and methodology to each individual component for this project is described in the following subsections.

## 3.1. Receipt Localization & Cropping

Receipt image localization involves determining the foreground versus the background for any given image. For this problem, this system assumes the following acceptable constraints:

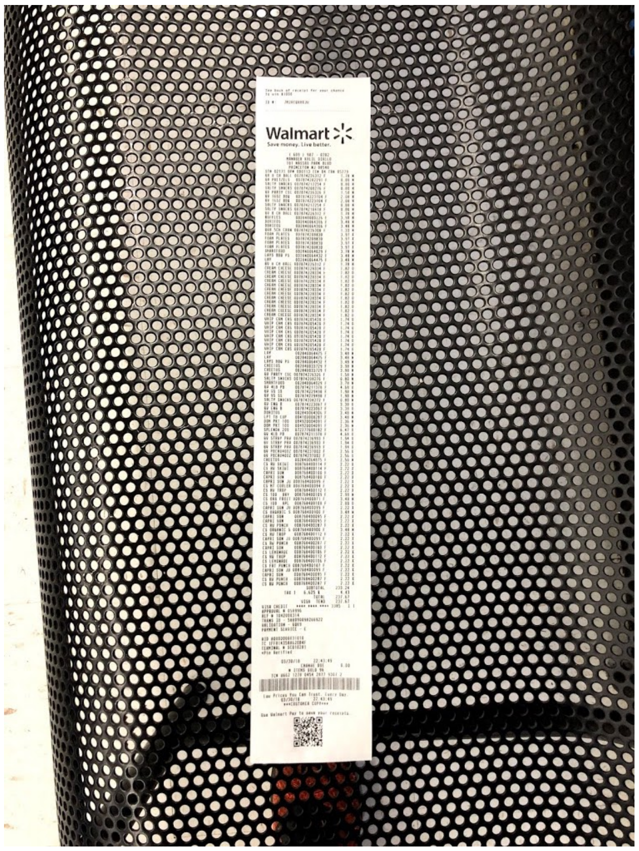
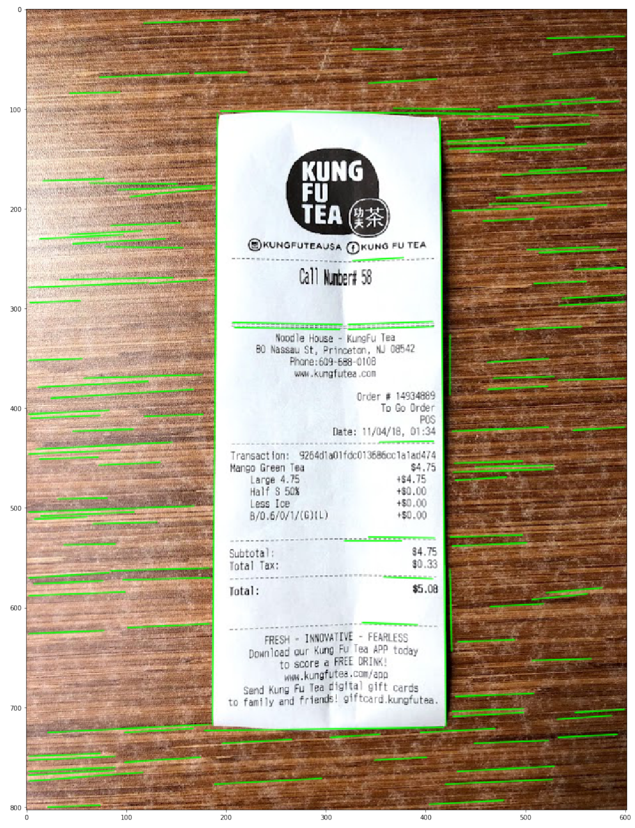


Figure 2. Sample Receipt Image



KUNG FU TEA

KUNGFUTEAUSA KUNG FU TEA

Call Number# 58

Noodle House - KungFu Tea
80 Nassau St, Princeton, NJ 08542
Phone:609-688-0108
www.kungfutea.com

Order # 14934889
To Go Order
POS
Date: 11/04/18, 01:34

Transaction: 9264d1a01fdc013686cc1a1ad474
Mango Green Tea $4.75
Large 4.75 +$4.75
Half S 50% +$0.00
Less Ice +$0.00
B/0.6/0/1/(G)(L) +$0.00

Subtotal: $4.75
Total Tax: $0.33

Total: $5.08

FRESH - INNOVATIVE - FEARLESS
Download our Kung Fu Tea APP today
to score a FREE DRINK!
www.kungfutea.com/app
Send Kung Fu Tea digital gift cards
to family and friends! giftcard.kungfutea.

Figure 3. Line Segment Detector Result for Receipt

- The receipt is rectangular or of a rectangular shape.
- The exists a distinguishable edge along the edge of the receipt that provides clear contrast with respect to the background.

Given the above constraints, processed images will result in convex quadrilaterals when projected onto a 2D space. Thus, our goal is to identify the four points that represent this quadrilateral and to then compute an affine transformation to project it onto a 2D space. One of the goals of this part of the system is to make it as robust as possible when identifying receipt images from backgrounds. The input data consisted of a wide variety of receipt sizes on an assortment of challenging, non-standard backgrounds. These backgrounds included objects such as park benches, multi-grained tables, and transparent glass surfaces.

Determining the edges of an image is an incredibly complex task when there exists a complex background. This system uses multiple, independent edge detection implementations that are combined using a statistical, weighted voting average in order to generate a final mask to determine the edge boundaries of the receipt image. The three different implementations used include a Line Segment Detector (LSD), Probabilistic Hough Transform processed through a Canny Edge Detector, and a deep learning aided Holistically-Nested Edge Detection (HED) model that leverages a fully convolutional neural network [8]. The combination of using these three processing techniques makes the cropping tool much more likely to identify the correct receipt area, and significantly boosts the accuracy of the overall system. This is due to the fact that each system has its own strengths and weaknesses with regards to edge detection; however, as the receipt image boundary is the most significant edge, it should appear in all three detection results.

The input image is first downscaled to a fixed size and transformed from the RGB color space to grayscale. This is done in order to improve algorithm performance as well as to increase the runtime speed. These down-sampled images are then parallelized into each of the following edge detectors described below.

### 3.1.1 Line Segment Detector (LSD)

The Line Segment Detector is a linear-time algorithm developed by Gioi, Jakubowicz, Morel, and Randall [6]. Essentially, it is a line extraction algorithm that uses Gaussian pyramids that are down-sampled and blurred in order to identify line support regions. These line support regions are then filtered using thresholding and region approximations in order to extract line segments independent of necessary parameter tuning.

The advantages of this algorithm is that it automatically controls for its own number of false detections through a thresholding method while simultaneously being able to provide sub-pixel accurate results [6]. Thus, LSD is able to identify all major, sharp edges within an image, regardless of the length or relevancy of the line. This provides a good baseline sampling of all available lines within the image, which are filtered by the weighted mask votes provided in each subsequent line process detector.

#### 3.1.2 Probabilistic Hough Transform

The Hough Transform is a technique used to determine features such as lines within an image. A binary image input, usually of edges from detectors such as the Canny Edge Detector, is used for this probabilistic method where possible lines matching the binary input are generated through a stochastic voting process and the randomly generated lines that overlap sufficiently enough with the binary input are subsequently chosen.

This method of image detection is used in this system because it is able to only detect the most significant edges for each given image. The grayscale input image is first convolved using a $5x5$ Gaussian kernel in order to remove noise from the image. This processed image is then run through a Canny Edge Detector with hysteresis thresholding and non-maximum suppression. This edge detector generates a binary map of edge locations which is then run through the Probabilistic Hough Transform to produce a map of predicted line segments. We then constrain the result space by filtering out lines that are shorter than a predefined minimum and are left with only leading edges.

#### 3.1.3 Holistically-Nested Edge Detection

Holistically-Nested Edge Detection is a novel edge detection algorithm by Xie that uses a deep learning model in order to resolve ambiguity over primary edges within a given image. It performs edge prediction through the use of a fully convolutional neural network and learns the significance of certain edges through training [8]. The model used for this project implements the architecture developed by Xie and uses the Berkeley Segmentation Dataset and Benchmark as the initial training data [5].

The advantages of using the HED machine learning model primarily stems from the fact that the machine learning model was trained to discriminate against flagging edges within the background of a given image. This method helped especially during time when input receipt images were especially noisy within the background and contained numerous straight lines. However, this algorithm requires a large amount of computational power in order to work and provides only marginal benefit in images where the receipt background was not extremely noisy.

### 3.2. Affine Transformation & Unwarping

Each line segment mask generated from the three independent edge detectors is combined into a single mask to determine the boundaries of the receipt image. This is done through a weighted probability distribution whereby the values in each mask are transformed to a value between 0 and 1 and then the mask is multiplied by a pre-computed weight specific to each edge detector. These weighted masks are then added together and their values quantized into a new binary mask where final lines are detected using a Probabilistic Hough Transform. These final lines generated by this transform represent the highest-confidence-predicted edges for the given input images, and is based on a combination of data provided by the three individual edge detectors that vote for each given line segment.

The lines are then segmented into vertical and horizontal lines using a $k$-means clustering algorithm that clusters based on the $\rho$ value of polar coordinates. Intersections between vertical lines and horizontal lines are then determined after the clustering is complete. Nearby intersecting points are then merged through averaging by using a kd-tree for efficient lookup. These points represent statistically likely points for corner points of the receipt image.

After likely corner points are generated through these statistical methods, the system assigns a box score to each combination of four points representing the edges of a proposed receipt image boundary within the final corner point list. This box score assigned to each combination of points is the likelihood of those points representing the actual receipt boundary edges. This score is computed by statistically weighting a combination of factors including the sharpness of each corner with respect to 90 degrees per corner, the distance the center of gravity of the points is to the center of the image, and the ratio of the area of the rectangular box to the total area of the image.

The combination of four points with the lowest box score is chosen as the edges of the receipt. These edges are then used to calculate an affine transformation matrix that will be used to crop the receipt image from its original form into an axis-aligned and in-plane image to be used for the optical character recognition system. This matrix is then applied to the original input image and then returned for use in later steps in this pipeline to extract text from the image.

### 3.3. Text Extraction Through OCR

The optical character recognition engine used for this implementation project is Tesseract OCR [**tesseract**]. Tesseract is a neural network enhanced optical character recognition engine that is maintained by Google, and is one

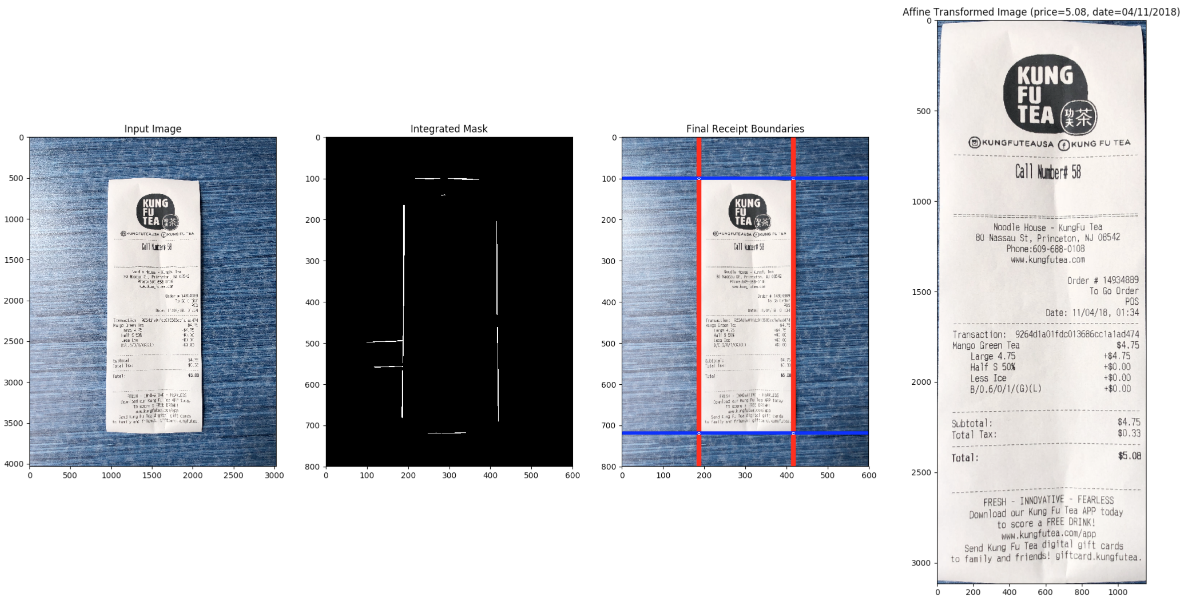


Figure 4. Basic Receipt Parsing with a Complex Background

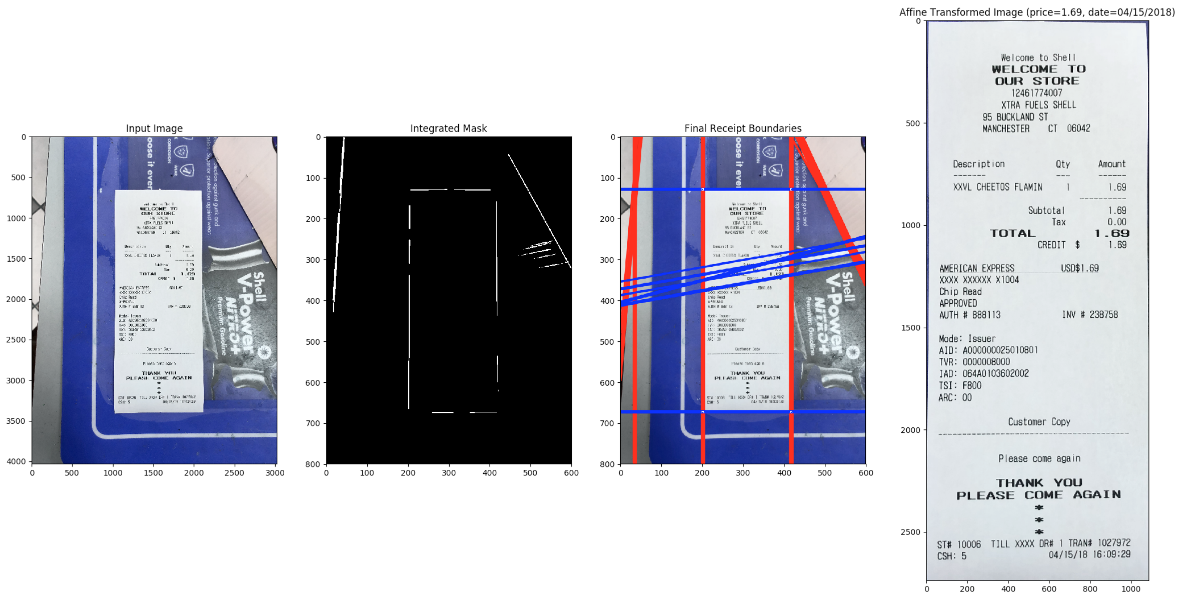


Figure 5. Receipt Parsing with a Noisy Background

of the leading open-source recognition systems available on the market.

Before images in the pipeline are inputted into Tesseract for recognition, a number of pre-processing and image enhancement steps are performed in order to boost the accuracy of the result outputted from the OCR engine. These steps include bilateral image filtering in order to remove noise from the image and image thresholding in order to quantize the image into a binarized version. Further enhancements performed on the input image include using algorithms such as the median blur algorithm.

The processed image is then passed into Tesseract, which then returns data including the text recognized, the bounding boxes of each recognized text grouping, and a confidence score with regards to how accurate the system predicts the recognition is. Each individual box is then filtered to remove clearly incorrect recognitions and recognitions where the system is not confident in.

The final bounding boxes for text contain individual groupings of text that the recognition system probabilis-

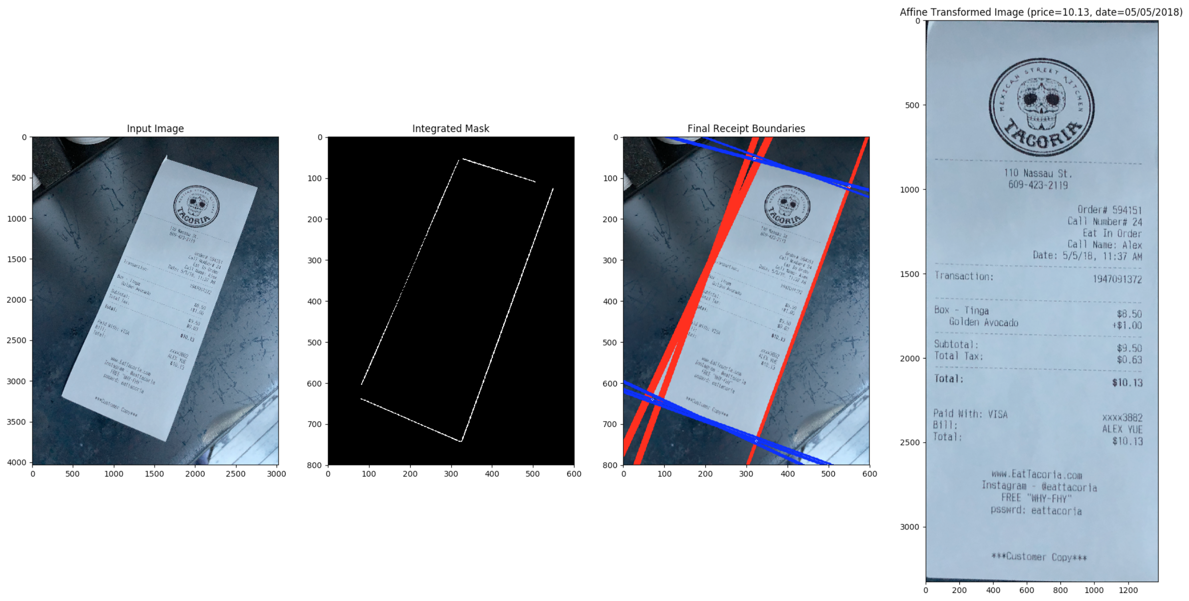


Figure 6. Receipt Parsing with an Affine Transformation

tically estimates should be grouped. This receipt parsing pipeline then runs a new set of steps in order to regroup semantically and geographically similar groups of text into larger bounding boxes. Such methods are completed by probabilistically selecting pairs of text dependent on the distance of two text bounding boxes. Given two text bounding boxes, boxes are merged if they contain lexically similar content such as address information or numerical information. For example, if two adjacent bounding boxes both contain numerical integers, the system will automatically merge both bounding boxes together and insert a decimal point given the statistically likely chance that the OCR system failed to recognize the period delineating the two. Once the final, merged bounding boxes are computed, they are passed onto the next stage of the pipeline to extract both date and price data from each receipt.

### 3.4. Semantic Meaning & Understanding

Understanding the semantic meaning of text is a very difficult problem, especially when there exists partial text recognition data from the OCR engine. To solve this challenge, this system uses a combination of natural language processing techniques such as tokenization, regular expression matching, and spatial search algorithms in order to identify data located on receipt images. I take advantage of commonalities found on many receipts such as shared location information for where common data such as total price and merchant name are typically located on receipt images. This semantic understanding system also takes advantage of keywords on the image that typically imply categories such as ”price”, ”total”, or ”amount” for the total price.

The two main categories that this system was built to parse were receipt dates and total prices. To parse a given OCR output, words are first tokenized using natural language processing libraries and then keywords for each given category identified. Once each keyword is identified, I ran a spatial search for all given text inputs nearest to the text input containing the keyword for insightful information. For example, for parsing pricing data, keywords such as ”price”, ”total”, and ”amount” are first identified from the OCR output. Then, for each given positive keyword match, a nearest-neighbor search is conducted to look for text bounding boxes containing pricing information. I then select the keyword-price pair for the bounding boxes that are the furthest down on the page, which usually represents the total price for a given receipt image.

Once the price and date information are found, this information is returned as output and a plot containing the individual images generated during the pipeline combined with the parsed information is displayed to the user.

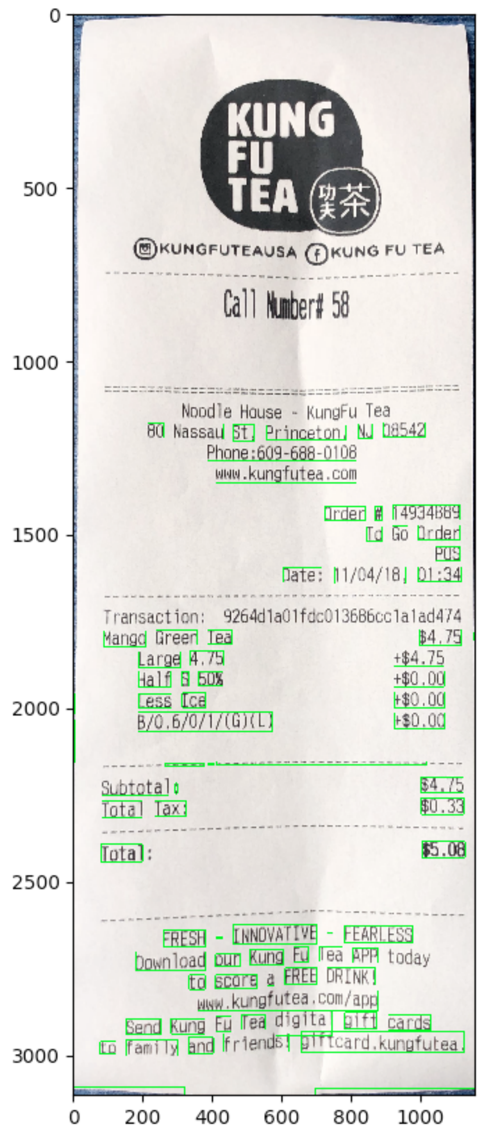
## 4. Documentation

### 4.1. Modules

The implementation for this project is primarily broken up into three distinct modules, all in charge of a unique subset of features:

- `launcher` - Responsible for integrating all modules into a single runtime and returning final output and render.

- `detector` - Contains all the primary logic for each


0
500
1000
1500
2000
2500
3000
0
200
400
600
800
1000

KUNG FU TEA 功夫茶

KUNGFUTEAUSA KUNG FU TEA

Call Number# 58

Noodle House - KungFu Tea
80 Nassau St, Princeton, NJ 08542
Phone:609-688-0108
www.kungfutea.com

Order # 14934889
To Go Order
POS
Date: 11/04/18, 01:34

Transaction: 9264d1a01fdc013686cc1a1ad474
Mango Green Tea $4.75
Large 4.75 +$4.75
Half S 50% +$0.00
Less Ice +$0.00
B/0.6/0/1/(G)(L) +$0.00

Subtotal: $4.75
Total Tax: $0.33

Total: $5.08

FRESH - INNOVATIVE - FEARLESS
Download our Kung Fu Tea APP today
to score a FREE DRINK!
www.kungfutea.com/app
Send Kung Fu Tea digital gift cards
to family and friends! giftcard.kungfutea.

Figure 7. Predicted Bounding Boxes for Receipt

of the edge detectors, and code to integrate and filter out the final detected edge segment results.

- `transform` - Responsible for taking in an edge mask and input image and returning an axis-aligned and in-plane cropped image.

- `recognition` - Responsible for running Tesseract OCR and parsing price and date data from the bounding box output from Tesseract.

### 4.2. Dependencies

This project could not have been finished without the work of numerous other modules that this system depends upon. Packages and scientific computing libraries used in this project include:

- OpenCV
- NumPy
- Caffe
- Tesseract
- matplotlib
- scikit-learn

## 5. Results

Included above and continuing into the next page are sample result outputs generated from this receipt recognition pipeline. Further results are also included that demonstrate the rotation invariant and other capabilities of the system.

## 6. Evaluation

This project is evaluated in two separate stages. First, there was a qualitative evaluation to test the performance of the receipt recognition, cropping, and parsing. Next, a quantitative evaluation was performed by running this system through a set of custom generated test receipt images to check for the accuracy of the system.

### 6.1. Qualitative Evaluation

From tests and run-throughs using this system pipeline, I felt that, subjectively, this system was relatively easy to use and fast at parsing information from receipt images, with most results being returned in sub-second intervals. However, the system does slow down when there are a lot of false positive lines segments that are identified during the line detection stage as the box score calculation algorithm needs to iterate through all possible combinations in order to determine the best combination of points available that represents the receipt image. This could be improved in the future through the finer tuning of the edge detection mask weights for each individual detector.

I evaluated the quality of the boundary detection for a sample of receipt images. The algorithm that was developed and implemented for detecting the receipt image boundaries performed relatively well as it was able to determine the image boundaries in the majority of the cases. This occurred in light of difficult test images where parts of the receipt were obscured or taken on challenging backgrounds such as multi-grained tables that would cause some line detectors to attempt to recognize the individual wood grains as line segments.

Through the use of multiple independent edge detectors that vote on the confidence of each proposed line segment, this system is much more robust to edge cases when compared to other document edge detectors.

### 6.2. Quantitative Evaluation

Data for use against a baseline system is difficult to come by as it does not exist. Thus, I have assembled a custom test dataset of 50 unique receipt images for use in evaluating the accuracy of this system. For comparison, these images were also provided to another receipt parsing system called Receipts by Wave.

Ultimately, I found that this system implementation outperformed the competing implementation. Furthermore, this system is much faster, with most results generated within 1 second compared to minutes for Receipts by Wave.

- This system was able to correctly identify prices from **36/50** of the receipt images.

- Receipts by Wave was able to correctly identify prices from **22/50** of the receipt images.

## 7. Discussion

The system developed over the course of this final project performed relatively well across a wide variety of test images. Given the subset of images tested, this pipeline performed optimally when the receipt images were crisp and separated by a solid, distinguishable background. The system would be able to identify data such as price with extremely high accuracy rates given these circumstances. In situations where the background was noisy, such as a marble countertop or on top of a table advertisement, the boundary detection statistical voting algorithm was able to identify the borders in most cases; however, when the boundary identification failed, the pipeline would not be able to function at all. One possible solution to allow the pipeline to still run if the boundary detection and cropping failed would be to align the receipt text to the horizontal axis using a classification algorithm and then run optical character recognition on the image to attempt to extract text from the entire image. During testing, this approach did not work nearly as well as the cropped version due to the additional noise added to the image from the background.

One other great strength of the system developed was that it was extremely fast in most circumstances. In timing tests of standard run-throughs of sample receipt images, the system would often provide a result within the second. Furthermore, the edge detection algorithm runs in near-real-time, allowing for further developments such as a visual overlay during the image capture process to act as an aid for the user.

## 8. Further Improvements

This receipt image parser provides a significant baseline platform for pulling pricing data off receipts. However, it is not to say that there cannot be further improvements on this system. One area of improvement that I will focus on implementing is training and inserting an end-to-end regional proposal network to identify areas of text and their bounding boxes within each receipt image. One of the shortfalls of Tesseract is that it is unable to handle font variations within an single image very well. There exist some receipt images within the training and testing datasets that have variations in font sizes, such as larger fonts for the logos and bolded font for the total price. Tesseract does not handle these type of receipt images very well and often fails to identify areas of text that have a font face or size different from the majority font used in each image. Adding a region proposal network that identifies text blocks would aid this by allowing this Tesseract to identify text individually per each bounding box rather than for the entire image at once.

Furthermore, another improvement that I would like to add to this project is to implement better methods to identify and correct errors that occur in the optical character recognition pipeline. Existing optical character recognition systems often will recognize characters within some words incorrectly. For example, the word "total" will sometimes be recognized at "tota1", where the letter "l" is substituted by the number "1". Training a LSTM that can take input text and autocorrect errors such as these would boost the accuracy in receipt identification and parsing significantly.

## References

[1] K. Barrett. Smartscan 101.

[2] L. ExpressExpense. Expressexpense's massive receipt database, 2016.

[3] M. Gjoreski, G. Zajkovski, A. Bogatinov, G. Madjarov, D. Gjorgjevikj, and H. Gjoreski. Optical character recognition applied on receipts printed in macedonian language. 04 2014.

[4] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1):1–20, Jan 2016.

[5] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[6] G. Randall, J. Jakubowicz, R. G. von Gioi, and J. Morel. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 32:722–732, 12 2008.

[7] B. Shi, X. Bai, and S. J. Belongie. Detecting oriented text in natural images by linking segments. *CoRR*, abs/1703.06520, 2017.

[8] S. ”Xie and Z. Tu. Holistically-nested edge detection. In *Proceedings of IEEE International Conference on Computer Vision*, 2015.

[9] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai. Multi-oriented text detection with fully convolutional networks. *CoRR*, abs/1604.04018, 2016.

## 9. Acknowledgements

Many thanks to Prof. Russakovsky who was the main advisor for this final project.

## 10. Honor Code

This paper represents my own work in accordance with University regulations.

/s/ Alex Yue