*International Journal of Computer Applications (0975 – 8887)*
*Volume 155 – No 2, December 2016*

# Identification of Receipts in a Multi-receipt Image using Spectral Clustering

Siddharth Garimella
Saint Francis High School
1885 Miramonte Ave
Mountain View,
CA 94040

## ABSTRACT

In order to submit expense reports, multiple receipts are often scanned on a single page and the scanned images are submitted along with the expense report in order to get expenses reimbursed. These scanned images are manually verified to check the validity of the claimed expenses. In this paper, a method is presented to isolate receipt segments in an image and use Optical Character Recognition (OCR) to identify receipt amounts, reducing validation time and effort. Scanned images are processed to find the contours of all high-contrast objects in receipts, including letters. Minimum bounding rectangles (MBRs) are found for each of the contours. Spectral clustering is used to group these MBRs in order to find receipt clusters which correspond to individual receipts. These are then processed with OCR to aid the user with validation.

## General Terms

Clustering, Pattern Recognition, Receipt Recognition, and Spectral Clustering Algorithm.

## Keywords

Clustering, Pattern Recognition, Receipt Recognition, and Spectral Clustering Algorithm.

## 1. INTRODUCTION

With the advent of cloud computing and SaaS (Software as a Service) applications, several enterprises now use expense report management cloud applications for processing expense reports. Expense reports contain a list of expenses and a set of scanned receipts. These receipts are manually validated in order to process expenses by reviewers and processors. Some of these applications try to use an OCR approach on the scanned receipts in order to identify the provider name and the total amount. However, these applications assume that each receipt is scanned individually. In reality, most of the expense reports are filed with multiple receipts scanned on a single page, often precluding the use of such applications without further processing the image. In this paper, an approach using Spectral clustering is presented to identify individual receipts in a multi-receipt image, extracting each one out for OCR processing. This approach greatly reduces the processing overhead of manually trying to locate a receipt in an image and matching the corresponding amount to the claimed amount of the receipt.

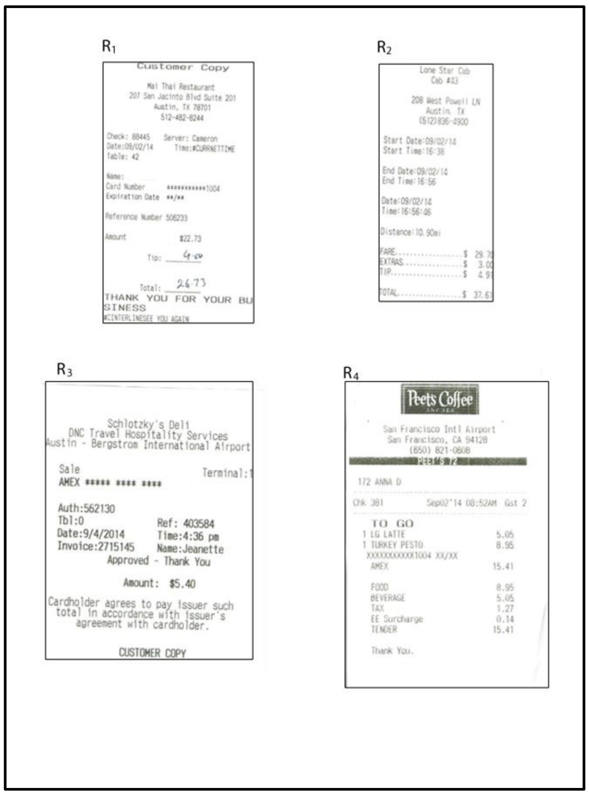
Several techniques are used to recognize text in an image using various Optical Character Recognition techniques [1]. These techniques assume that text is located in one area (block) of the image. However, if a single image has multiple blocks of text, it needs to be pre-processed and the text blocks need to be extracted in order to identify text in a meaningful way. In the context of current expense report management applications, each of the text blocks correspond to a receipt that is presented as a proof for that expense report. If the background image has high contrast with the receipt, it is relatively easy to identify the receipt's contours. However, most scanned images have white backgrounds, complicating this problem. With new HD scanners and cameras, high resolution images help to recognize the text in each of the receipts with great accuracy once they are identified and separated from the rest of the image.

In this paper, an approach is presented to pre-process the image and use Spectral clustering to identify receipt blocks in scanned images with a white background. Identified receipts are further processed by an OCR module in order to identify information that aids in faster expense report reviews and approvals.

## 2. PRE PROCESSING OF SCANNED IMAGE

A scanned image is expected to have one or more receipts in it and it is assumed that the number of receipts, which is provided in the expense report, is known. If the background color provides proper contrast, it is relatively easy to identify the receipt contours. In this paper, it is assumed that the scanned image background is white, as most scanners produce an image with a white background, as shown in the example in Figure 1. For clarity, all four receipts are labeled with minimum bounding rectangles on a scanned image, I, with width, W, and height, H.

Let us consider that each image has n receipts $R_1, R_2, R_3 ... R_n$ that are included in it. Each receipt, i, occupies a rectangular area with width, $w_i$, and height, $h_i$, and has lower left hand corner coordinates, $(x_i, y_i)$. It is expected that receipts occupy a rectangular area and are typically aligned with the scanned image and also don't overlap with each other. Each receipt has a set of objects that include letters, logos, signatures, etc. They all are located in a specific rectangular area. The example, shown in Figure 1, contains 4 receipts $R_1, R_2, R_3, R_4$.



$R_1$

Customer Copy

Mai Thai Restaurant
207 San Jacinto Blvd Suite 201
Austin, TX 78701
512-482-8244

Check: 88445 Server: Cameron
Date:09/02/14 Time:#CURRNETTIME
Table: 42

Name:
Card Number ***********1004
Expiration Date **/**

Reference Number 506233

Amount $22.73

Tip: 4.00

Total: 26.73
THANK YOU FOR YOUR BU
SINESS
#CINTERLINESEE YOU AGAIN

$R_2$

Lone Star Cab
Cab #43

208 West Powell LN
Austin, TX
(512)836-4900

Start Date:09/02/14
Start Time:16:38

End Date:09/02/14
End Time:16:56

Date:09/02/14
Time:16:56:46

Distance:10.90mi

```
FARE.................$ 29.70
EXTRAS...............$  3.00
TIP..................$  4.91

TOTAL................$ 37.61
```

$R_3$

Schlotzky's Deli
DNC Travel Hospitality Services
Austin - Bergstrom International Airport

Sale Terminal:1
AMEX ***** **** ****

Auth:562130
Tbl:0 Ref: 403584
Date:9/4/2014 Time:4:36 pm
Invoice:2715145 Name:Jeanette
Approved - Thank You

Amount: $5.40

Cardholder agrees to pay issuer such total in accordance with issuer's agreement with cardholder.

CUSTOMER COPY

$R_4$

Peet's Coffee

San Francisco Intl Airport
San Francisco, CA 94128
(650) 821-0608
PEET'S 72

172 ANNA D

Chk 381 Sep02'14 08:52AM Gst 2

```
TO GO
1 LG LATTE                5.05
1 TURKEY PESTO            8.95
XXXXXXXXXXXX1004 XX/XX
 AMEX                    15.41

 FOOD                     8.95
 BEVERAGE                 5.05
 TAX                      1.27
 EE Surcharge             0.14
 TENDER                  15.41

 Thank You.
```

**Figure 1: A Scanned Multi-receipt Image**

One simple approach is to consider all pixels as data points and use general clustering algorithm to find clusters of receipts. This approach suffers from two problems. The first is that the large data set (the number of pixels on the scanned image) is difficult to cluster and the second is that clusters that are obtained may not be rectangular. For example, if the page is scanned at 144 DPI (Dots Per Inch), a letter sized, 8.5 by 11 inch scanned image contains 1,938,816 pixels. Computing a similarity matrix, approximately 2 trillion entries, and identifying clusters will take up unusual amount of memory and computing power. In order to achieve a more computationally efficient and effective method, it is important to pre-process the image and generate a meaningful dataset before further processing with clustering. The preprocessing steps involve the following:

1. Convert the colored image into its grayscale equivalent
2. Blur the image using a median filtering algorithm
3. Run Canny Edge algorithm to highlight all contours in the image
4. Find MBRs (Minimum Bounding Rectangles) for all the contours found in the image
5. Remove any rectangles with insignificant areas
6. Remove any rectangle that is contained in another rectangle

The above steps help to create a small data set that is easy to process and cluster. In the first step, any colored image is converted into grayscale. Most of the images are corrupted by noises due to faded receipts, scanner resolution, and background noise. There are linear or nonlinear filter methods to reduce noise. The median filter, a nonlinear filter, has been widely used in digital image processing because of its good edge retaining characteristics while reducing impulse noise

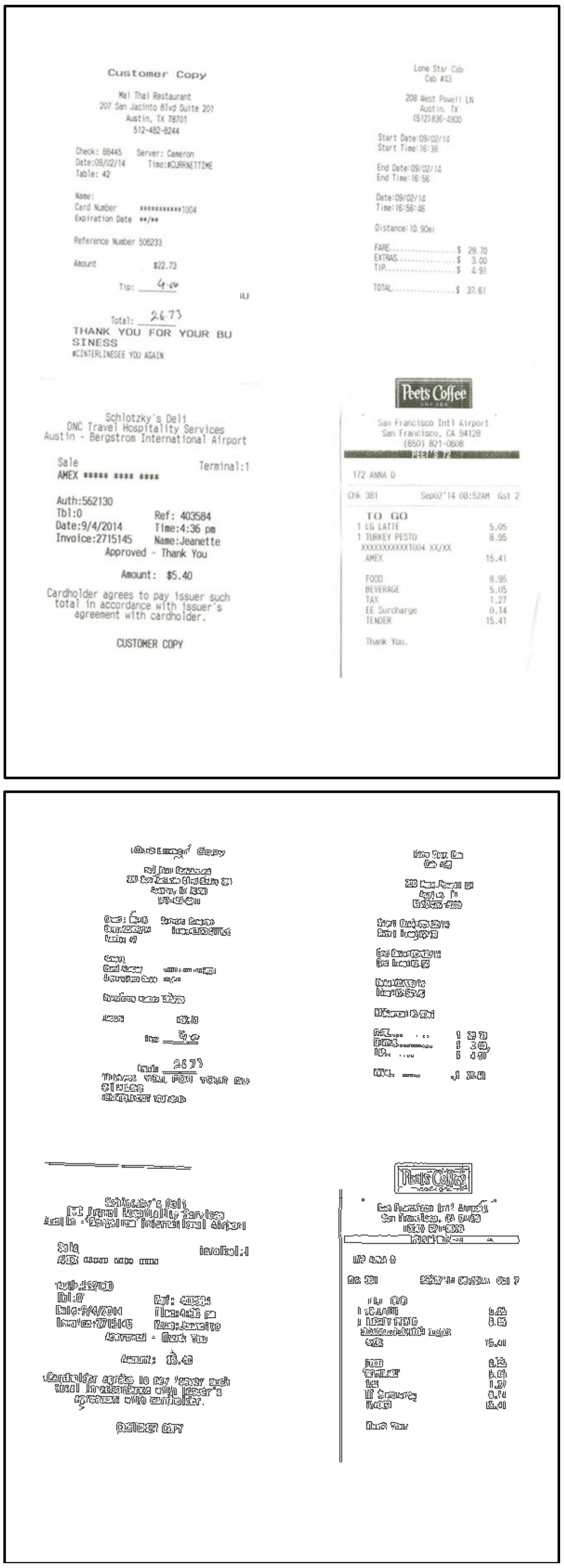


**Figure 2: Image output at the end of Step 2 and Step 3**

ability [2]. The median filter is a rank-order filter. A pixel value of the digital image is replaced by the median value of the neighborhood pixels. The medium value of the ranked neighborhood pixels is used to replace the noisy value. The median filtering output calculated using $I_1(x, y) =$

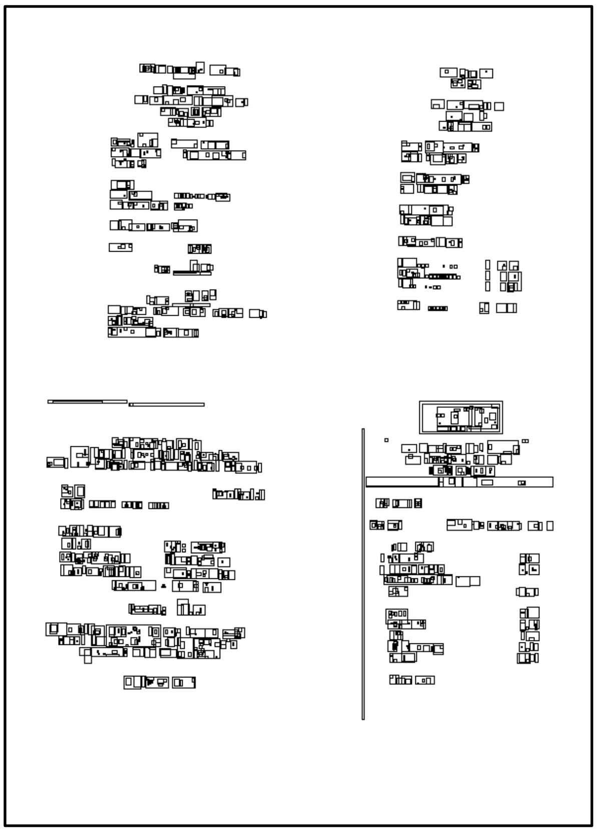
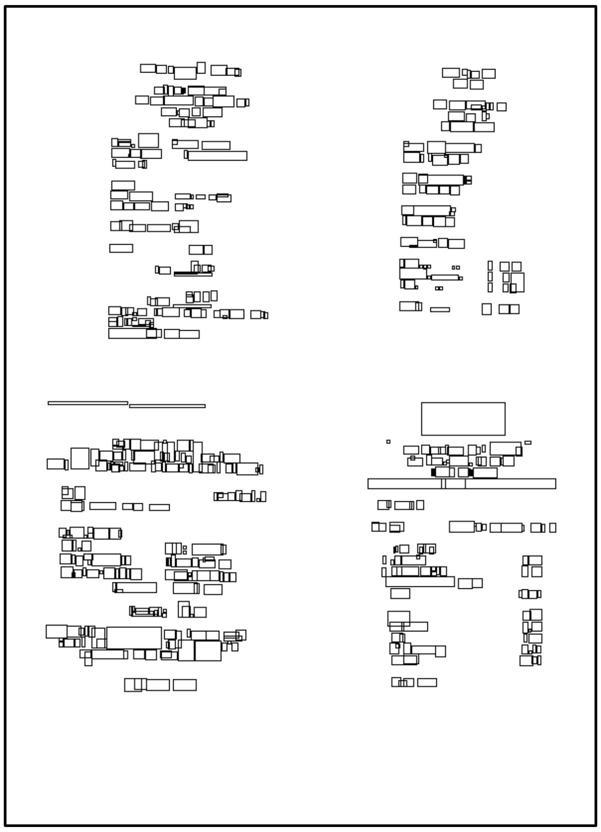



**Figure 3: Image output at the end of Step 4, Step 5 and Step 6**

$\text{Medium}\{I(x-i, y-j), (i,j) \in M\}$, where $I(x,y), I_1(x,y)$ are the pixel values at $(x,y)$ of the original image and the output image respectively, M is a two-dimensional mask and the mask size is $m \times m$. Typically, the value of m is odd such as 3 or 5. In the second step, a median filter is applied on the image to reduce noise in the image. Figure 2 shows the result after applying medium filter for $m = 3$ in this Step 2.

In the third step, all edges or contours are detected. The purpose of edge detection is to help reduce the amount of data in an image by using the detected contours to process the image instead of whole image. There are several algorithms to detect edges that have been proposed in the literature Canny Edge detection algorithm developed by Jon F. Canny[3] is the standard for edge detection methods still used in research. In this paper, Canny method is used to identify all edges or contours in the image. By processing image $I_1$with Canny edge detection algorithm, it will obtain a set of contours or edges, $E = \{e_1, e_2, e_3, e_4, \ldots, e_P\}$ where P is the number of contours identified in the image and each edge has a set of points, $e_i = \{c_1, c_2, c_3, c_4, \ldots, c_{iq}\}$. Figure 2 shows the output of identified contours using the Canny Edge detection algorithm.

In the fourth step, a minimum bounding rectangle, $R_i$, is identified for each of the detected contours, $e_i$. This is done by finding minimum and maximum x and y values for all points in the contour. At the end of this step, it will have a set of rectangles representing the data. For properly scanned receipts, ideally, each of the characters in the receipt should have a rectangle. Sometimes, receipt edges will show as long, thin rectangles with insignificant areas. These rectangles often mislead clustering algorithms. In fifth step, all long, more than 144 pixels, and thin, less than 10 pixels, rectangles are removed from the data. Note that these threshold values depend on the resolution of the scan, DPI (Dots per Inch). In the last step, all rectangles that are fully contained in other rectangle(s) are also removed. This step reduces data further. At the end of this step, it will have a minimal set of rectangles in the two dimensional space. In Figure 3, the results after Steps 4, 5 and 6 on the original scanned image shown in Figure 1 are presented.

In the next section, a Spectral clustering algorithm is used on this set of rectangles to identify receipt rectangle areas in the scanned image.

## 3. CLUSTERING DATA

After preprocessing the scanned image, the data comprises of a set of P rectangles that need to be clustered. As mentioned before, the number of clusters to be identified based on the data is known in the expense report.

Clustering helps to find pattern associations by forming groups of patterns such that a pattern in a group is more similar to other patterns in the same group when compared to patterns in other groups. Many clustering approaches have been proposed in the literature [4]. Many of the conventional clustering approaches are numerical clustering approaches which assume that patterns are points in a ddimensional space and perform clustering by defining a (dis)similarity measure [5]. In the current context, clustering helps to group rectangles that belong to a receipt into a single cluster. Once the clusters are identified, the corresponding receipts can be extracted from the image for OCR (Optical Character Recognition) processing. In order to employ a clustering algorithm, a distance measure needs to be defined between two data items. In case of point data in d dimensional, distance measure can be a simple Euclidean distance. However, rectangles are two dimensional objects with width and height. The distance between two rectangles cannot be a simple Euclidean distance as sometimes they overlap and sometimes they may be touching on one of the axis. The following approach is used to find the distance between two rectangles $r_1(x1, y1, x2, y2)$ and $r_2(x1, y1, x2, y2)$. Each rectangle has $(x1, y1)$ as upper left hand corner and $(x2, y2)$ as lower right hand corner. Here $r_1.x1$refers to x1 coordinate of $r_1$. The distance, $Distance(r_1,r_2)$, between $r_1$ and $r_2$ is computed using the following distance function presented in Figure 4.

```
Distance(r1, r2)

Begin

    If r1 and r2 are overlapping then return 0.0
    Else
    Begin
        // Find the mostLeft and mostRight rectangles
        Rectangle mostLeft, mostRight, upper, lower;
        Double xDiff, yDiff;

        If (r1.x1 < r2.x1) then mostLeft = r1 else mostLeft = r2;
        If (r1.x1 > r2.x1) then mostRight = r1 else mostRight = r2;

        If (mostLeft.x1 == mostRight.x1) xDiff = 0.0
        Else xDiff = max(0.0, mostRight.x1 – mostLeft.x2);
        // Find the upper and lower rectangles
        If (r1.y1 < r2.y1) then upper = r1;
        Else upper = r2;
        If (r1.y1 > r2.y1) then lower = r1;
        Else lower = r2;
        If (upper.y1 == lower.y1) yDiff = 0.0
        Else yDiff = max(0.0, lower.y1 – upper.y2);
        // Find the distance
        Return sqrt(xDiff * xDiff + yDiff *yDiff);
    End
End
```

**Figure 4: Distance measure between two rectangles.**

A similarity matrix, $S_{P\times P}$, can be calculated by subtracting the distance between two rectangles from the maximum of the distances' between any two rectangles in the data set. Note that P represents the total number of rectangles in the data set.

Several clustering algorithms were evaluated including K-means and found that Spectral clustering is suitable for finding proper clusters in this type of data set. The Spectral clustering method is easy to implement and is performant for sparse data sets that contains thousands of data items. In addition, it considers data clustering as a graph partitioning problem without any assumption on cluster structure. For more information theory behind Spectral clustering, please refer to [6]. Spectral clustering has two stages. In the first stage, it forms an associated Laplacian matrix [7], computes eigenvalues and eigenvectors, and maps points to lower dimensional representation based on eigenvectors. In the second stage, it assigns data items to one or more classes based on new data representation. In this paper, the SMILE[8] (Statistical Machine Intelligence and Learning Engine) library (Smile) is used to perform Spectral clustering using the distance measure presented in Figure 4. Once clusters are obtained, Minimum Bounding Rectangles for each of the clusters are computed using the data items in the corresponding cluster. Each of these rectangles identifies a receipt in the scanned image.

Figure 5 presents all the identified clusters found and their minimum bounding rectangles using Spectral clustering algorithm. All four receipt image sections, $RI_1$, $RI_2$, $RI_3$, and $RI_4$ are extracted from the image.

## 4. POST PROCESSING IMAGES WITH OCR

There has been a lot of research done in the area of Optical Character Recognition (OCR) [1]. There are several approaches and software available to identify text from

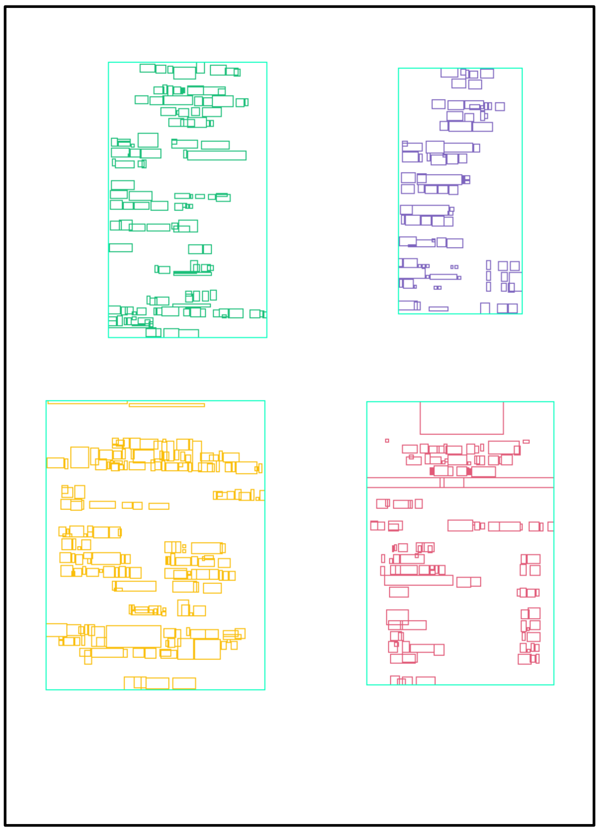


**Figure 5: Clusters found by Spectral clustering algorithm**

scanned images. One of the popular libraries is Google Vision API. Each of the receipt image sections are post processed using Google Cloud Vision APIs [9] to identify the total amounts paid by the expense report submitter. The proposed approach was accurately able to identify the amounts and match them to the amounts specified in the main expense report by the user.

The proposed approach was executed on a number of (105) scanned expense report images. In 93% of the cases, Spectral clustering was able find proper clusters. In addition, the approach was tested on different image resolutions including 72 DPI, 144 DPI, 200 DPI and 300 DPI. Image resolution didn't change the results. However, when the receipts are overlapping or very close to each other, it fails to find accurate clusters. Additional heuristics such as default minimum and maximum widths of receipts can be used to aid this clustering algorithm to identify proper clusters.

## 5. CONCLUSIONS

In this paper, an approach is presented and has been used in order to develop receipt processing software for identifying different receipts in a scanned multi-receipt image for automated recognition of amounts in the receipts. This greatly aids in the faster processing of expense reports that have scanned copies of the receipts. A scanned multi-receipt image is preprocessed to create a manageable data set that contains a set of rectangles representing objects and text in a receipt. Using Spectral clustering, all the identified minimum bounding rectangles are clustered and the associated receipt image sections are obtained. Once the receipt sections are identified, OCR library is executed on each of the sections to identify the expensed amounts in the receipts that aid in the faster approval of expense report. Future research can be done in this area to address the problem with overlapping and/or closely placed receipts in the scanned image. Additional heuristics such typical receipt width can be used to separate clusters containing multiple receipts if they are closer.

## 6. ACKNOWLEDGMENTS

I would like to thank my managers at Infinira Software for providing research opportunity, guidance and support for my research work on developing a library for receipt recognition software.

## 7. REFERENCES

[1] Mori, S., Suen, C. Y., and Yamamoto, K. 1992. Historical Review of OCR Research and Development, IEEE Proceedings, vol. 80, no. 7, 1029-1058.

[2] Huang, T. S. and Tang, G. T. 1979. A fast two-dimensional median filtering algorithm, IEEE Trans Acoustics, Speech, and Signal Processing, vol.27, no. 1, 13- 18.

[3] John Canny. 1986. A computational approach to edge detection. Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8(6):679–698.

[4] Jain, A. K. and Dubes, R. C. 1988. Algorithms for Clustering Data. Englewood Cliffs, NJ: Prentice-Hall.

[5] Jain, A. K., Murty, M. N., and Flynn, P. J. 1999. Data clustering: A review. ACM Computing Surveys, vol. 31(3), 264-323.

[6] Ng, A. Y., Jordan, M. I., and Weiss, Y. 2002. On spectral clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems 14, volume 14, 849-856.

[7] Mikhail Belkin and Partha Niyogi. 2003. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. Neural Computation, vol. 15, 1373-1396

[8] Smile – Statistical Machine Intelligence and Learning Engine (http://haifengl.github.io/smile/).

[9] Ray Smith. 2007. Tesseract OCR Engine, https://tesseract-ocr.googlecode.com/files/TesseractOSCON.pdf, Google, Inc.