IJRESM
International Journal of Research in Engineering, Science and Management
Volume-2, Issue-4, April-2019
www.ijresm.com | ISSN (Online): 2581-5792

OCR to Read Credit/Debit Card Details to Autofill Forms on Payment Portals

Pranav Raka[1], Sakshi Agrawal[2], Kunal Kolhe[3], Aishwarya Karad[4], R. V. Pujeri[5], Anita Thengade[6], Uma Pujeri[7]

[1,2,3,4]*Student, Department of Computer Science and Engineering, MIT College of Engineering, Pune, India*
[5,6,7]*Professor, Department of Computer Science and Engineering, MIT College of Engineering, Pune, India*

***Abstract*: In the world that is rapidly moving towards automation, why should authenticating transactions fall behind. With OCR we can auto fill credit card information to make it easy for use and avoid the hassle of retyping. Optical Character Recognition (OCR) is the mechanical or electronic conversion of images of typed, handwritten or printed text into machine-encoded text. We aim to read text from the credit/debit card and extract the details from it and auto-fill this information into the form to reduce human effort.**

***Keywords*: Optical Character Recognition, Machine Learning, Human Computer Interaction**

1. Introduction

A. Problem Definition

The Optical Character Recognition (OCR) is a wide domain of research in Artificial Intelligence, Pattern Recognition and computer vision. OCR is a technique of texts that are handwritten or digitized pictures so that they could be stored and searched more efficiently and correctly. While study of decades and enhancement in this technology, machines are still nowhere near to human's capabilities of analysis. The objective of an OCR method is recognition of scripts same as humans do. The advancements in pattern recognition has broadly accelerated due to the many emerging applications such as document classification, computer vision, data mining, shape recognition etc. OCR is a complex problem because of the variety of languages, fonts and styles in which text can be written, and the complex rules of languages. Hence, techniques from different visions computer science such as image processing are employed to address different challenges. Nowadays, many organizations are depending on OCR systems to eliminate the human interactions for better performance and efficiency and accuracy. Optical Character Recognition is a system that provides a full alphanumeric recognition of printed or handwritten characters at electronic speed by simply scanning the document. Documents are scanned using a scanner and are given to the OCR systems which recognizes the characters in the scanned documents and converts them into ASCII data.

B. Requirement Analysis

One of the more tedious steps in online shopping is filling out credit card forms during checkout. Using this system, the user can simply hold up the card to the camera and the form will be auto filled. There is a concern to users who store their card details on the websites. The website can get hacked and the card details can be leaked to malicious users. In case, users want to store their card details after the first "read", they can use the currently existing technology where the card details are encrypted and stored. Using this technology, users who are illiterate or users who are extremely new to online shopping can proceed with ease without having to go through with the complicated part of the shopping experience.

We intend to create a prototype system which runs the OCR on the server and the user access it from their browser at the client side.

C. Expected Results

The expected result of our system is that it will capture the image of the card and successfully recognize the 16-digit card number with at least 90% accuracy and expiry date. The system will recognize the card type using the first 4 digits of the card number. The image captured will follow the minimum quality so that the characters will be recognized correctly. After recognition of card number and expiry date it will proceed to the checkout page for facial recognition.

D. Features of OCR

The OCR system provides the following features:

- No more retyping
- Saves Time
- No Human Mistakes
- Ease of Use

E. Literature Survey

Computers understand alphanumeric characters as ASCII code which are typed on a keyboard. However, computers cannot distinguish characters and words from scanned images of paper documents. To solve this problem characters must first be converted to their ASCII equivalents before they can be recognized as readable text. Optical character recognition system (OCR) allows us to convert a document into electronic text. Character recognition is not a new problem but its roots can be traced back to systems before the inventions of computers. The earliest OCR systems were not computers but mechanical devices that were able to recognize characters, however it had certain disadvantages of very slow speed and low accuracy rate. J. Rainbow, in 1954, devised a machine that can read uppercase typewritten English characters, one per minute. The early OCR systems lacked the capability of providing high speed and greater accuracy due to which not much research work was done during 60's and 70's.OCR-A and OCR-B were developed by ANSI and EMCA in 1970, that provided comparatively acceptable recognition rates. In last few years great research work has been done in the field of OCR which has eventually lead to emergence of document image analysis, multi-lingual, handwritten and omni-font OCRs. However, the ability of machine to read text is still far below the human. Hence, more and more research is being done to provide greater accuracy and speed. Various methods have been proposed to increase the accuracy of optical character recognizers. The current challenge is to develop robust methods that remove as much as possible the typographical and noise restrictions while maintaining rates similar to those provided by limited-font commercial machines. Thus, current active research areas in OCR include handwriting recognition, and also the printed typewritten version of non-Roman scripts. There has not been availability of any open source or commercial software available for complex languages like Urdu or Sindhi etc.

Tesseract is in the top three OCR engines in terms of character accuracy in 1995.It is most widely used recent OCR technology. It is available for Linux, Windows and MAC OS.

Tesseract up to version 2 could only accept TIFF images of simple one-column text as inputs. Since version 3.00 Tesseract has supported output text formatting, OCR positional information and page-layout analysis. Support for a number of new image formats was added using the Leptonica library.

The initial versions of Tesseract could only recognize English-language text. Tesseract v2 added six additional Western languages (French, Italian, German, Spanish, Brazilian Portuguese, Dutch). Version 3 extended language support significantly to include ideographic (Chinese & Japanese) and right-to-left (e.g. Arabic, Hebrew) languages, as well as many more scripts. New languages included Arabic, Bulgarian, Catalan, Chinese (Simplified and Traditional), Croatian, Czech, Danish, German (Fraktur script), Greek, Finnish, Hebrew, Hindi, Hungarian, Indonesian, Japanese, Korean, Latvian, Lithuanian, Norwegian, Polish, Portuguese, Romanian, Russian, Serbian, Slovak (standard and Fraktur script), Slovenian, Swedish, Tagalog, Tamil, Thai, Turkish, Ukrainian and Vietnamese. Currently, we are using Tesseract 4.0 with LSTMs.

2. Optical character recognition

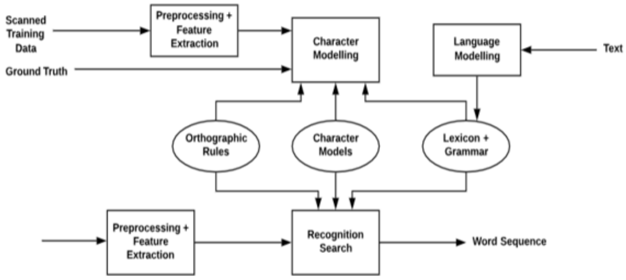

Fig. 1. OCR system

Optical character recognition or i.e. OCR, is a method of translating a scanned image into text format. Initially when a page is scanned, it is normally stored as a bit-mapped file in TIF format. When the image is displayed on the screen, we can read it, but to the computer, it is just a series of black and white dots. The computer does not recognize any "words" on the image. This part of processing is handled by OCR. OCR looks at each line of the image and attempts to determine if the black and white dots represent a particular letter or number.

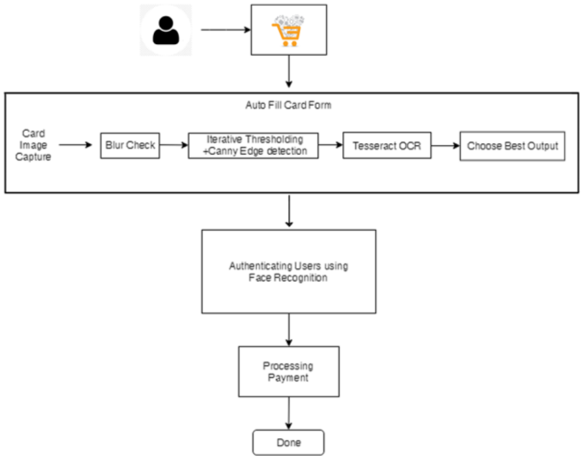

Fig. 2. Our system pipeline

A. *Scanning*

Initially the required document will be scanned by the scanner. Scanning speed will be determined by the quality of the scanner machines, color, paper quality, cleanness, weights, right setting of the OCR system.

A bounding box of the exact dimensions of a credit/debit card is displayed. The aspect ratio of the card is 17:11. This bounding box indicates to the user how close to the camera the card must be held. While capturing the image, the image had to be flipped around the vertical axis so that user sees a "mirror reflection" and to avoid confusion regarding left and right movement.

B. *Pre-processing*

As soon as the document is scanned it is preprocessed. The image is firstly converted to a gray scale image and then to a binary image. This process of conversion is called as Digitization.

In pre-processing, a convolution operation is done on the image using a Laplacian filter. The variance of this operation is used to determine whether the image is blurry or not.

To get the best result, we must get a training data of about 80-100 equally blurred and unblurred images and find the boundary value of the variance using a simple State Vector Machine(SVM) or a logistic regression. And use that value as a boundary to determine whether image is blurry or not.

For the initial stage, we have manually determined based on about 5-10 images what the variance should be to determine the sharpness of the image.

The image is the converted into a grayscale image using functions from the openCV2 library. Based on testing using various thresholding for the image (To have all the values in the image to be only 1 or 0), Gaussian thresholding, or thresholding after gamma correction were not giving serviceable results.

So we performed manual thresholding. The manual threshold was changed from 70 to 100 by increments of 10 (min = 0, max = 255) and 4 different images were obtained.

Another technique we used was using a canny filter after performing a gaussian blur.

These images were then fed directly to Tesseract for performing Optical Character Recognition.

C. Character Extraction

The pre-processed image is searched to find out every single character in the image. This recognized characters are used for comparison with the stored data. Here we use the in-built tesseract function to extract the characters from the card.

D. Recognition

It compares the value of the interpreted image with all the templates which are preloaded into the system. Once the comparison is done, the recognized characters are displayed in text format.

The string returned from the tesseract OCR was further processed by removing all characters except:
numbers,”/”,”\n”,” “(space) using the regular expression:
reg=r"(\d)|(/)|(\n)|( )"

Each character in the output from tesseract was matched with this regular expression and the ones who matched, were taken for further processing. Each character in the output from tesseract was matched with this regular expression and the ones who matched, were taken for further processing. From all the outputs obtained, we calculated the most number of numeric characters and the string with the most numeric characters was declared as the best string.

Then, for parsing the best string, we implemented the following 2 methods:

Method 1:

First split the best string by “\n”, then split all the obtained string using “ “.

Now, the strings which contain 4 characters are considered part of the card number.

Method 2:

First split the best string by \n, then give the longest string as best output.

3. Math models

To detect text and more importantly in this case numbers from the image captured we use pre-processing techniques. First, we have the Laplacian filter which is used to detect edges and the sharpness of the image.

The variance of this image tells us if the image is blurry or not and removes noise from the image.

The Laplacian L(x,y) of an image with pixel intensity values I(x,y) is given by:

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

L(x,y)= (delta^2 I)/(delta x^2) + (delta^2 I)/(delta y^2)

This is computed using a convolution filter. The most commonly use kernels for this operation are these given below.

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

4. Results

Based on testing using various thresholding for the im-age (To have all the values in the image to be only 1 or 0), Gaussian thresholding, or thresholding after gamma correction were not giving serviceable results. So we performed manual thresholding. The manual threshold was changed from 70 to 90 by increments of 5 (min = 0, max = 255) and 4 different images were obtained. These images were then fed directly to Tesseract for performing Optical Character Recognition. The best out of the results for all the images is used as the output to be given to the Tesseract module.

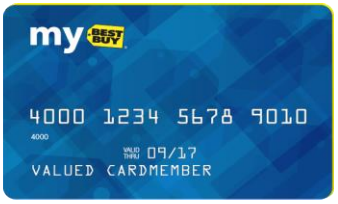

Fig. 3. Input image

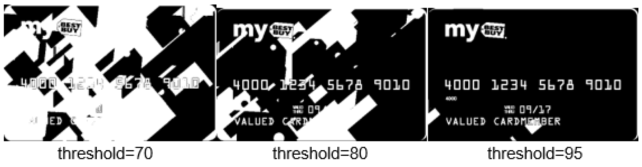
threshold=70 threshold=80 threshold=95

Fig. 4. Output

5. Applications

OCR can contribute to a large number of applications such as:

In early days, it was used for mail sorting, bank cheque reading and verification of signatures, it can also be used by organizations that undergo automated form processing in places where a huge number of data is available in printed form. Processing utility bills, passport validation, pen computing and number plate recognition etc. It can also be helpful to blind and visually impaired people to read text.

6. Conclusion

After performing preprocessing, we are getting an accuracy of 87.5%.

If we train the Tesseract and add a new layer to it and train it, we will get better results. We first tried template matching but since it yielded very low results we chose to discard that approach.

References

[1] R. Lienhart, and A. Wernicke, “Localizing and segmenting text in images and videos”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 12 (4), 2002, pp. 256-268.

[2] D. Doermann, J. Liang, and H. Li, “Progress in Camera-Based Document Image Analysis”, Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR 2003), 2003.

[3] X. Chen, J. Yang, J. Zhang and A. Waibel, “Automatic detection and recognition of signs from natural scenes”, IEEE Transactions on Image Processing, vol. 13 (1), 2004, pp. 87-99.

[4] S. Marinai, M. Gori, and G. Soda, “Artificial neural networks for document analysis and recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27 (1), 2005, pp. 23-35.

[5] Refael C. Gonzalez, Richard E.Woods and Steven L. Eddins, Digital Image Processing using MATLAB, Prentice Hall, 2004.

[6] Rosebrock, A. (2017, July). Pyimage search. https://www.pyimagesearch.com/2017/07/17/credit-card-ocr-with-opencv-and-python/.

[7] Yann LeCun, Leon Bozou, Yoshua Bengio, and Patrick Huffner, “Gradient-Based Learning Applied to Document Recognition,” proc. of the IEEE, November 1998.