Text Retrieval from Scanned Forms Using Optical Character Recognition

Vaishali Aggarwal, Sourabh Jajoria and Apoorvi Sood

Abstract This paper investigates the use of image processing techniques and machine learning algorithm of logistic regression to extract text from scanned forms. Conversion of printed or handwritten documents into digital modifiable text is a tedious task and requires a lot of human effort. In order to automate this task, we apply the machine learning algorithm of logistic regression. The main components of this system are (i) text detection from the scanned document and (ii) character recognition of the individual characters in the detected text. In order to complete these tasks, we firstly use the image processing techniques to do line segmentation, character segmentation, and then ultimately character recognition. The character recognition is done by a one-vs-all classifier which is trained using the training data set and learns the parameters with the help of this data set. Once the classifier has learned the parameters, it could identify a total of 39 characters which include capital English alphabets, numerals, and a few symbols.

Keywords OCR · Logistic regression · Segmentation · Classifier

1 Introduction

In the real world, organizations or even individuals are faced with the problem where they have hard copies of certain documents and there is a need to convert the text in these documents to a digital form where operations like searching,

V. Aggarwal (✉) · S. Jajoria · A. Sood
Netaji Subhas Institute of Technology, Sector 3, Dwarka, New Delhi, India
e-mail: vaishalia809@gmail.com

S. Jajoria
e-mail: sourabhjajoria@gmail.com

A. Sood
e-mail: soodapoorvi@yahoo.com

S. Urooj and J. Virmani (eds.), *Sensors and Image Processing*, Advances in Intelligent Systems and Computing 651, https://doi.org/10.1007/978-981-10-6614-6_21

modification, insertion, [1] and others could be performed on the text. A system that could do this task is called an optical character recognition system (OCR) [2]. One important application of this system is to create a database from the information in filled forms which would otherwise be a very cumbersome task if done manually. The aim of the proposed model was to build an OCR that takes the scanned image of the document and accomplishes the aforementioned task. The OCR is required to be as accurate as possible. The most important step which determines the accuracy of the prediction is character recognition [3]. A one-vs-all classifier which uses a machine learning algorithm logistic regression [4–6] was used to build the OCR.

The idea behind one-vs-all classifier is that there were several classes which needed to be recognized. A training data set with objects of a number of known classes was used to train the classifier. The training data was composed of several entries where each entry had the values for a set of features along with the label for that entry. The classes in the case of an OCR were the set of characters that the system could recognize. For a given input character that had to be recognized, probabilities were computed using each of the classifier. Out of the resultant probabilities, the class for which the probability was the highest was given as the predicted class.

Section 2 describes OCR system design, Sect. 3 discusses the results of the experiments performed and a comparative study of the proposed model with other commercial OCR tools, and Sect. 4 concludes the paper providing some insight into the future applications of the proposed work.

2 OCR System Design

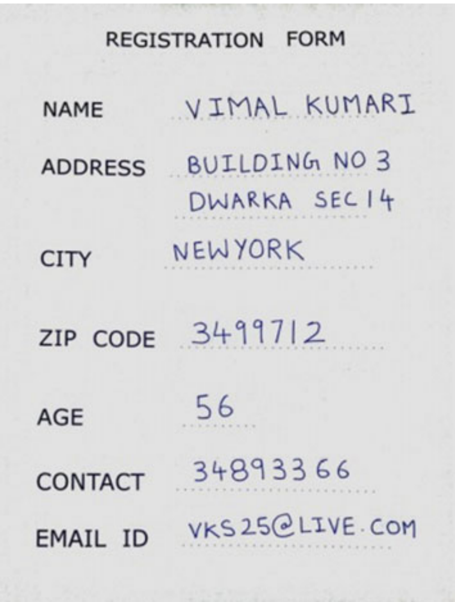
The main modules of the system [7–9] are described in this section.

a. Image acquiring, b. Pre-Processing, c. Line Segmentation, d. Character Segmentation, and e. Character recognition

2.1 Image Acquisition

In order to process a document [10, 11], firstly an image of the document was needed. The document in this case was a form which had only the text part and no images in it. The image had to be of good quality since the accuracy of the OCR depends highly on it. Noise had to be as low as possible. A scanner is the best device for this purpose. If the scanned image still contained some noise, then it had to be manually filtered out from it (Fig. 1).

Fig. 1 Scanned Form



REGISTRATION FORM

NAME VIMAL KUMARI

ADDRESS BUILDING NO 3
DWARKA SEC 14

CITY NEWYORK

ZIP CODE 3499712

AGE 56

CONTACT 348933 66

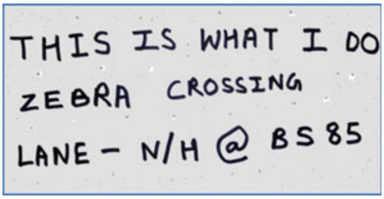
EMAIL ID VKS25@LIVE.COM

2.2 *Pre-Processing*

Accuracy of an OCR system depends on text pre-processing and segmentation algorithms [12]. Pre-Processing can be further divided into the following steps:

- *The first step was to convert the colored image into a grayscale image, which consisted of shades of gray.*
- *Grayscale image was then converted into black and white image to filter out the unwanted noise.*
- *Small objects from binary image* [13] *were removed, i.e., objects that had fewer pixels than a cut-off value were removed from the black and white image. This step proved to be very efficient in the removal of noise* (Fig. 2, 3, and 4).

Fig. 2 Original Image



THIS IS WHAT I DO
ZEBRA CROSSING
LANE – N/H @ BS 85

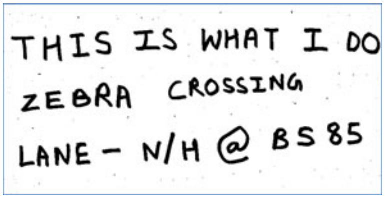


Fig. 3 Grayscale Image

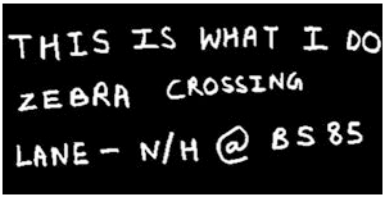


Fig. 4 Filtered Image

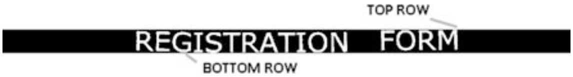


Fig. 5 Cropped Line

2.3 *Line Segmentation*

The image of the scanned document was to be processed line by line. Line Segmentation is the process of extraction of lines of text from the form and working on each line separately to facilitate character recognition in it. An important point to note down here is that the processing was based on inverted black and white images where the background is black with a pixel value of 0 and the written text is white with a pixel value of 1 [14, 15]. Steps applied for this purpose:

1. Starting from the top of the image, the row of pixels where the sum of pixel values was not zero was searched. This marked the beginning of the first line in the document.
2. The image was scanned till the sum of pixel values in that particular row was greater than zero.
3. The row where the sum of pixel values was zero marked the bottom of the current line. The aforementioned top and bottom pixel rows were then used to crop out a line from the image which was then used for character recognition.
4. The above three steps were repeated for the remaining image to get the remaining lines (Fig. 5).

2.4 *Character Segmentation*

Character Segmentation was used to obtain individual characters from the form field line obtained from line segmentation [16]. Each character was enclosed in a bounding box. To do this correctly, there needed to be some space between

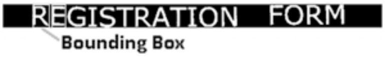


Fig. 6 Characters in Bounding Box

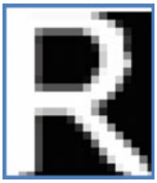


Fig. 7 After top, bottom, left, and right cropping

characters. Now the bounded image of character was further cropped from all four sides of the box. This was done by scanning for the first line with non-zero pixel value sum from top, bottom, left, and right of the bounding box (Fig. 6 and 7).

2.5 *Character Recognition*

A one-vs-all logistic regression model was used to recognize the characters. Since there were 39 different characters that the system could recognize, 39 different classifiers were trained, one for each character.

A dataset of around 2000 handwritten and printed characters was used. This data was then used to train the classifier. Each element in the training data was a 24 * 42 pixel image of a printed or handwritten character (Fig. 8).

2.5.1 Logistic Regression

The logistic regression hypothesis is defined as:

$$h_\theta(x) = g\left(\theta^T x\right) \tag{1}$$

Here, g is the sigmoid function which is defined as:

$$g(z) = \frac{1}{1 + e^{-z}} \tag{2}$$

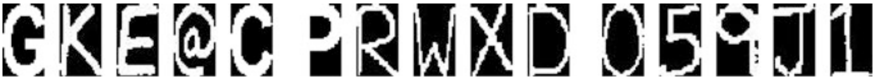


Fig. 8 Training Data samples

Θ is the parameter matrix whose value is different for all 39 classifiers. The dimensions of Θ for each classifier are 1 * (n + 1), where n is the number of features.

Logistic regression cost function:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log\left(h_\theta\left(x^{(i)}\right)\right) - (1-y^{(i)})\log(1-h_\theta\left(x^{(i)}\right))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2 \quad (3)$$

Where m is the total number of rows in the training data, x specifies each row of the training data and y specifies the label value for that particular row. The value of y varies from 1 to 39, one for each character. The second term in the above equation is the regularization term. Regularization was used to avoid the problems of over-fitting and under fitting. λ is the regularization constant.

Here, the cost function J(Θ) was to be optimized by changing the parameter Θ. For that, gradient of the cost function was calculated for each column in the Θ vector.

The gradient is a vector whose j^{th} element is:

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - (y^{(i)})x_j^{(i)}\right) + \frac{\lambda}{m}\theta_j \quad for\ 1 \le j \le n \quad (4)$$

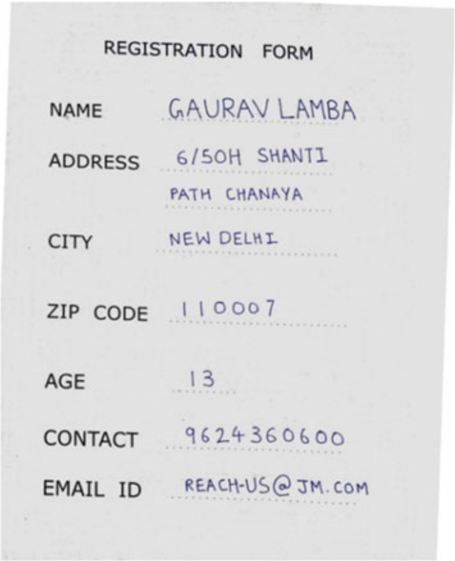
$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - (y^{(i)}t)x_j^{(i)}\right) + \frac{\lambda}{m}\theta_j \quad for\ 1 \le j \le n \quad (5)$$

This gradient function helped in determining the value of the parameter Θ for which the cost function was minimum. Thus, by finding the value of Θ, one-vs-all classifier was trained. In order to recognize a character, the probability that it belongs to each class using trained classifiers for each of the 39 characters was calculated. The label (1–39) for which the value of the probability was the highest was the character predicted by the one-vs-all classifier.

3 Results

The system was tested on several scanned images of forms which constituted the testing set. These forms contained printed fields and some handwritten data filled using ball-point or gel pen. The image was scanned using Microsoft Office Lens.

Figure 10 *shows the data stored in digital text format from recognition performed on* Fig. 9 *using our system.* The recognition rate for text on forms is between 85% and 90%, which is similar to other systems used in character recognition. There is still scope for improvement in recognizing similar characters like ‘0’ and ‘O’, ‘H’ and ‘B’, and ‘1’ and ‘I’. Recognition rate depends a lot on these two factors:



REGISTRATION FORM

NAME GAURAV LAMBA

ADDRESS 6/50H SHANTI
PATH CHANAYA

CITY NEW DELHI

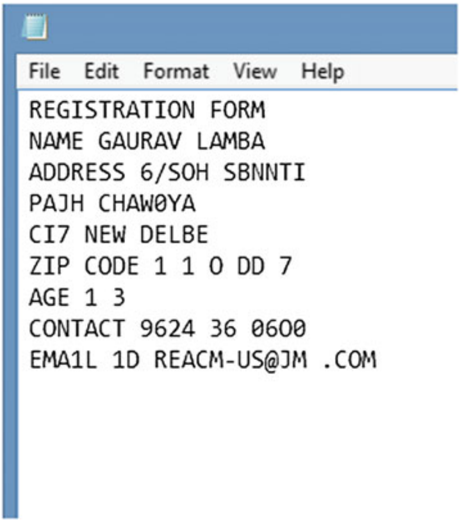
ZIP CODE 110007

AGE 13

CONTACT 9624360600

EMAIL ID REACH-US@JM.COM

Fig. 9 Input Form



```
REGISTRATION FORM
NAME GAURAV LAMBA
ADDRESS 6/SOH SBNNTI
PAJH CHAW0YA
CI7 NEW DELBE
ZIP CODE 1 1 0 DD 7
AGE 1 3
CONTACT 9624 36 0600
EMA1L 1D REACM-US@JM .COM
```

Fig. 10 Output

a. Quality of scanned forms and
b. Handwriting used on forms (Fig. 11).

The accuracy of character recognition was computed at different stages of development, and it was found that the accuracy of proposed OCR was proportional

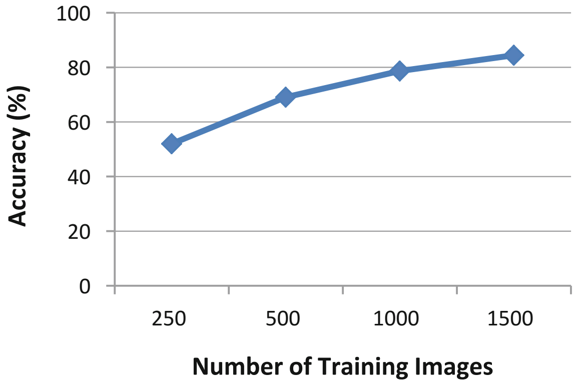


Fig. 11 Character recognition accuracy as a function of number of training images

Table 1 OCR Comparison

Input Image		Accuracy of OCR					
Image number	Number of characters	Suggested system	ABBYY FineReader 12	I2OCR	Tesseract	Cuneiform	Free OCR to word
1.	129	88.3	87.6	34.9	19.3	76.7	41.1
2.	128	93.7	85.2	42.2	57.0	58.6	64.4
3.	130	92.3	77.7	62.3	65.4	50.8	58.5
4.	125	79.8	73.4	52.6	59.9	51.2	55.3
5.	128	83.6	69.2	62.3	67.3	63.7	61.0
Average accuracy		87.4	78.6	50.9	53.8	60.2	56.1

to the number of training images in the training set at initial stages. The proposed system was also compared with some well-known free OCR tools like Tesseract, Cuneiform, ABBYY FineReader 12(Trial Version), I2OCR, and Free OCR to Word. They provide good accuracy and speed. Many other OCR tools are proprietary and paid. The accuracy of some tools was very less on the test set and hence they are not mentioned here. The experiment was carried out on a computer with Intel Core i3 1.7 GHZ CPU, 2 GB RAM, and Windows 8 OS. All tools were tested on the same scanned form images which were in the test set.

Table 1 compares all the relevant tools to extract text from scanned forms. The result of OCR processing shows that ABBYY FineReader 12 provides 78.6%, I2OCR provides 50.9%, Tesseract provides 53.8%, Cuneiform provides 60.2%, and Free OCR to Word provides 56.1% of average accuracy in the test set, whereas the suggested system provides an average accuracy of 87.4%.

4 Conclusion and Future Work

We tried to build a text recognition system that could be used on manually filled forms containing handwritten as well as printed English alphabets written in uppercase, numerals, and some special characters. There are numerous important domains of application of this system like library registration, banking forms, and many more. We have shown that text can be recognized with a reasonable accuracy using simple image processing techniques and logistic regression algorithm for multiclass classification of characters. It works well for a variety of documents having no prior knowledge of character size, font color, and document layout.

Future work in this system consists of modifying this approach to recognize lowercase letters and texts with different orientation. This system works for few special characters, so there is scope to add more characters. The machine learning algorithm used for classification is logistic regression; as an improvement, a more complex algorithm like neural networks can be used for better accuracy. Thus, one would hopefully obtain a more accurate recognition of filled forms to apply this system in practical use.

References

1. Mohammad, F., et al. (IJCSIT) International Journal of Computer Science and Information Technologies, **5**(2), 2088–2090 (2014)
2. Wolf, C., Jolion, M.J., Chassaing, F.: Text localization, enhancement and binarization in multimedia documents. In: International conference on pattern recognition, pp. 1037–1040, 2002
3. Kahan, S.T., Pavlidis, T., Baird, W.: “On recognition of printed characters of any font and size”, IEEE transactions of pattern recognition and machine intelligence, pami-91987, pp. 274–285
4. Hosmer, D., Lemeshow, S.: Applied logistic regression, 2nd edn. Wiley, New York (2000)
5. Harrell, F.: Regression Modeling Strategies: With Applications To Linear Models, Logistic Regression, and Survival Analysis. Springer, New York (2001)
6. Logistic regression and artificial neural network classification models: A methodology review. J. Biomed. Inform. **35**, 352–359 (2002)
7. Jain, A.K., Bhattacharjee, S.: Text segmentation using Gabor filters for automatic document processing. Mach. Vis. Appl. **5**(5), 169–184 (1992)
8. An embedded application for degraded text recognition. EURASIP J. Adv. Signal Process. **2005**(13), 2127–2135 (2005)
9. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Blobworld: Image segmentation using expectation-maximization and its application to image querying. IEEE Trans. Pattern Anal. Mach. Intell. **24**(8), 1026–1038 (2002)
10. Wolf, C., Jolion, J-M. Extraction and Recognition of Artificial Text in Multimedia Documents. http://rfv.insalyon.fr/wolf/papers/tr-rfv-2002-01.pdf
11. Doermann, D., Liang, J., Li, H.: Progress in camera-based document image analysis, in Proc. 7th IEEE International Conference on Document Analysis and Recognition (ICDAR’03), vol. 1, pp. 606–617, Aug 2003
12. Optical character recognition by open source OCR tool tesseract: a case study, Int. J. Comp. App. **55**(10), 0975–8887 Oct 2012

13. Matsuo, K., Ueda, K., Michio, U.: Extraction of character string from scene image by binarizing local target area. Transaction of The Institute of Electrical Engineers of Japan, 122-C(2), 232–241, Feb 2002
14. Gao, J., Yang, J.: An adaptive algorithm for text detection from natural scenes, in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01), vol. 2, pp. 84–89, Kauai, Hawaii, USA, 2001
15. Sobottka, K., Bunke, H., Kronenberg, H.: Identification of text on colored book and journal covers, International Conference on Document Analysis and Recognition 57–63 1999
16. Chen, X., Yuille,A.: Detecting and reading text in natural scenes. In: Computer Vision and Pattern Recognition, vol. 2 (2004)