International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 2 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT1951152

# Text Recognition from Document Images

**Revanth Yenugudhati*, Suresh Babu Papanaboina, Suryatej Vasireddy, Yaswanth Seelam**
Computer Science and Engineering, Vasireddy Venkatadri Institute of Technology, Guntur, Andhra Pradesh, India

## ABSTRACT

The objective is the development of effective reading skills in machines. After reading the text and comprehending the meaning, it would know itself and according to the program, it would implement the instructions. The current investigation presents an algorithm and software which detects, recognizes text and character with specific protocol in a image and programs itself according to the text. Technological advances in image processing have acquainted us with character recognition and many such related technologies, which have proved to be a milestone. However, even years after the invention of these technologies we have not been able to achieve a technology by which machine can read, interpret and act according to the instructions and even update their database if required. Here's an attempt to make this reality. Machine replication of human functions, like reading, is a long-awaited dream. However, over the last five decades, machine reading has transformed from a dream to reality. Text detection and character recognition known as Optical Character Recognition (OCR) has become one of the most successful applications of technology in the field of pattern recognition and artificial intelligence. Numerous commercial systems for OCR exist for a variety of applications.
**Keywords:** Image Processing, Text recognition, text detection, Optical Character Recognition, Text Detection and Character Recognition.

## I. INTRODUCTION

In recent years, text recognition from images and videos gained popular interest with advancement in pattern recognition and computer vision technology. The significance of semantic or high-level text data present in an image is that it can easily describe an image with good clarity and can be extracted using low-level features like colour, texture etc., which in turn varies with language, font, style and background, thus making the task of text extraction a challenging one. Recognition of text is yet another challenge for the researchers as low-resolution text with small fonts may be present in an image or video with complex or textured background.

The aim of the proposed method is to detect and recognize the text from the document images. The first stage of this method is text detection and the second stage deals with recognition of the detected text characters. The proposed algorithm begins with pre-processing of the input image. The given image is converted into binary image. The binary image is segmented using Line detection and Character detection algorithm. An OCR technique is used for recognizing the characters.

## II. METHODS AND MATERIAL

This section describes the details of our proposed scheme. The process of text recognition is divided

into three major parts: (A) Pre-processing (B) Segmentation (C)Text Recognition.

(A) Pre-processing of image

In pre-processing, a colour image is converted into a grayscale image. The binarization of the grayscale image is again divided into Variance Computation Broken Edges Linking and Adaptive Thresholding. Text connected components have a few basic properties based on which they can be separated from the image – they have a distinct boundary that separates it from the background non-text components. These basic features of text regions are exploited to classify the image into text and non-text clusters.

Generally, edge detection techniques are applied to find the boundary of a text. There are many edge detection techniques but the basic problem is that they are highly susceptible to noise. Edge detection works by computing the gradient of the image and then finding the high gradient line segments. Noise causes high gradient values and thus contribute to faulty edges. Here we have used another method to detect the boundary. If we traverse along a text boundary, the boundary points will produce a large variance of the grayscale values of the neighbouring pixels. If we slide a 5x5 window over the entire image it will produce high variance in the text boundary regions and comparatively low value in non-boundary portions of the image. Even noise points do not contribute to a significant increase in variance magnitude. Thus, this method is immune to noise and also effectively detects boundary regions with high accuracy. So, first we convert the input grayscale image into variance image. A higher value of the variance around a pixel makes the pixel darker while a pixel around which this variance is low, will tend to be more white. This is similar to the gradient map in edge detection techniques. Canny edge detector uses the gradient map to find the edges. Here, we use the principle of Canny edge detector on the variance image to find the edge map that we call the boundary lines.

The boundary lines (EG1) come from the variance image and partially gives the boundaries of the text regions. But there are discontinuities in the boundaries owing to low variance regions. Thus to complete the boundary description we take the help of canny edge detector. We find the edge (EG2) of the input gray image using canny edge detector with a very small threshold so that even the small edges become visible. Out of all the edges detected by the Canny edge detector we keep only those edges which are connected with the boundary. To do this we perform logical OR operation on EG1 and EG2 and keep the output in L. Now we form a new matrix (F1) where we store all the connected components (CC's) of L which are connected with the boundary pixels of EG1. Now, after performing this step, there might be small discontinuities in the boundary. We connect them using morphological bridge operation and store this in a map named boundary. Thus we get the complete boundary map of image which is almost noise free and can be used to binarize the image.

We obtain the horizontal run in a row of the boundary image from one boundary pixel to the next boundary pixel. We move from left to right for each row over the entire boundary image. Then we compute the mean value over the two $3 \times 3$ neighbourhoods around these two boundary pixels (starting and ending). This is the place where there are major changes in the gray scale values. So we compute the mean of these eighteen pixels and then compare all the pixels inside these two neighbourhoods with this mean. If it has a higher value than the mean, we assign 1 to the corresponding pixel; otherwise, we assign 0 to it. Then the same procedure is applied in vertical direction. Using this we create two matrices, namely, horizontal run matrix and vertical run matrix. We

then find the final matrix by logical AND on both horizontal run matrix and vertical run matrix. This gives us the final binarized image.

(B) Segmentation

The segmentation is the most important process in text recognition. Segmentation is done to make the separation between the individual characters of an image. Segmentation is one of the most important phases in this project. The performance of of this project is depending on segmentation. Segmentation subdivides an image into its constituent regions or objects. Basically in segmentation, we try to extract basic constituent of the script, which are certainly characters. This is needed because our classifier recognizes these characters only. In this project, we perform the segmentation of character from image by applying Line detection and Character detection algorithm which are discuss as follows:

Algorithm: Line detection
Step 1: Start scanning the image horizontally from the topmost left corner row by row.
Step 2: If any black pixel is encountered in a row make the row status as '0'.
Step 3: If no black pixel in encountered in a row while tracing it then marks the row status as '1'.
Step 4: By counting and following the total numbers of continuous '0' from row status vector number and position of lines can be obtained
Algorithm: Character detection
Step 1: Take a single line under consideration.
Step 2: Start scanning the image vertically from the topmost left corner column by column.
Step3: If any black pixel is encountered in a column mark the column status to '0'.
Step 4: If no black pixel in encountered in a column while tracing it then marks the column status as '1'.
Step 5: By counting and following the total numbers of continuous '0' from column status vector number and position of lines can be obtained.

(C) Text Recognition

The text recognition process is again divided into two part They are:

1) Feature Extraction:
Feature extraction is the process to retrieve the most important data from the raw data. The most important data means that's on the basis of that's the characters can be represented accurately. To store the different features of a character, the different classes are made. There are many techniques used for feature extraction like Principle Component Analysis (PCA), Linear Discriminate Analysis (LDA), Independent Component Analysis (ICA), Chain Code (CC), zoning, Gradient Based features, Histogram etc. In this we use matrix feature extraction method. In this method first we convert the image to binary matrix i.e. black and white image convert to matrix form, it may look like as shown in figure 2. in the above figure text image is converted in to the matrix of 0's and 1's.from this matrix data we was extract text character line by line and word by word by using above segmentation method. After that segmented characters data are normalized and store in fixed dimension as a feature of that character.

2) Classification:
The classification is the process of identifying each character and assigning to it the correct character class, so that texts in images are converted in to computer understandable form. This process used extracted feature of text image for classification i.e. input to this stage is output of the feature extraction process. Classifiers compare the input feature with stored pattern and find out best matching class for input. There are many techniques used for classification such as Artificial Neural Network (ANN), Template Matching, Support Vector Matching (SVM) etc. In this we use Artificial Neural Network (ANN) for classification because neural network can get itself trained automatically on the

basis of efficient tools for learning large databases and examples. This approach is non-algorithmic and trainable. There are the different types of neural networks which can be used for the classification from which we used Kohonen neural network.

## III. RESULTS AND DISCUSSION

Experiments are conducted on document images with two different datasets. Sample results are presented alongside quantitative results. Precision and recall metrics are used to measure the performance. These metrics are mainly based on true positive and false positive regions found. Also to evaluate the contributions of these two different metrics f is used, which is the harmonic mean of precision and recall as given below equation.

$$f=2/((1/precision)+(1/recall)$$

For testing we have taken two datasets RVL-CDIP dataset and PRImA dataset respectively. A subset from PRImA dataset have been used. First, to be able to see unprocessed document image using OCR. In this case, performance for document images is quite low since engine failed at finding any text in a document image.

Quantitative performance of our algorithm are given in tables. Compared to pure OCR, our algorithm is superior for all metrics on both datasets. On the other hand, another a method using sample points for character recognition by manwatkar obtained 0.67% for f metric with 0.72% precision and 0.60% recall on ICDAR dataset. Our algorithm performs better in text detection and those were shown in the following metrics precision, recall and f metric by testing on two different datasets.

TABLE I: CDIP DATASET RESULTS FOR DOCUMENT IMAGES

| | Precision | Recall | F |
|---|---|---|---|
| Only OCR | 0.573 | 0.537 | 0.553 |
| Proposed method | 0.827 | 0.580 | 0.520 |

TABLE II: PRIMA DATASET RESULTS FOR DOCUMENT IMAGES

| | Precision | Recall | F |
|---|---|---|---|
| Only OCR | 0.441 | 0.821 | 0.588 |
| Proposed method | 0.670 | 0.584 | 0.605 |

After detection of the text, text recognition is the next step and recognition is also performed well by the designed algorithm. The results for text recognition for the given sets are given below table.

TABLE III: ANALYSIS OF TEXT RECOGNITION RESULTS

| Methods | Character confidence | Character accuracy rate(%) | Error rate(%) |
|---|---|---|---|
| MSER and SWT based recognition | 40.6 | 45.9 | 54.1 |
| Poisson method | 45.8 | 53.6 | 46.4 |
| Proposed method | 90.10 | 92.42 | 4.68 |

## IV.CONCLUSION

In this paper, we proposed a novel system for detection and recognition of text, which combines several image processing components. It can be seen in experimental results, when different image processing methods in text detection and recognition are gathered together, promising results can be obtained without leveraging any benefit of deep

learning paradigm, only by image processing techniques, satisfactory performance is reached.

This work provides an improved text recognition methodology on document images. It uses both the edge and variance information of the input image. The proposed solution is not very sensitive to image color, text font, skew and perspective variation. The proposed method is effective in terms of low contrast and noisy text-based images. Experiments on proposed system outperforms precision and F-measure. Our future plan is to design a text recognition system with more accuracy in detecting and recognizing the text.

## V. REFERENCES

[1]. N. Otsu, "A threshold selection method from gray-level histograms," IEEE Trans. System Man and Cybernetics, vol. 9, no. 1, pp. 377-393, 1979.

[2]. W. Niblack, An Introduction to digital image processing. Prentice Hall, Englewood Cliffs, 1986.

[3]. Keechul Jung, Kwang In Kim and Anil K. Jain, "Text information extraction in images and video: a survey," The journal of the Pattern Recognition society, 2004

[4]. Victor Wu, Raghavan Manmatha, and Edward M. Riseman, "TextFinder: An Automatic System to Detect and Recognize Text in Images," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21, No. 11, November 1999.

[5]. Y. Zhong, K. Karu, and A.K. Jain, "Locating Text in Complex Color Images," Pattern Recognition, vol. 28, no. 10, pp. 1,523-1,536, Oct. 1995.

[6]. Optical Character Recognition (OCR). AIM, Inc. 634 Alpha Drive Pittsburgh, Pa 15238-2802, USA.

[7]. Yang, Jufeng, Kai Wang, Jiaofeng Li, Jiao Jiao, and Jing Xu, "A fast adaptive binarization method for complex scene images," 19th IEEE International Conference on Image Processing (ICIP), 2012.

[8]. Shrey Dutta, Naveen Sankaran, PramodSankar K., C.V. Jawahar, "Robust Recognition of Degraded Documents Using Character N-Grams," IEEE, 2012.

[9]. Gur, Eran, and ZeevZelavsky, "Retrieval of Rashi Semia Cursive Handwriting via Fuzzy Logic," IEEE International Conference on Frontiers in Handwriting Recognition (ICFHR), 2012.

**Cite this article as :**

Revanth Yenugudhati, Suresh Babu Papanaboina, Suryatej Vasireddy, Yaswanth Seelam, "Text Recognition from Document Images", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 2, pp. 75-79, March-April 2019. Available at doi : https://doi.org/10.32628/CSEIT1951152

Journal URL : http://ijsrcseit.com/CSEIT1951152