Read Textual features in Images and convert to Editable form by extended use of Artificial Neural Networks, Deep learning and Maximally Stable Extremal Region techniques

S. Joseph Gladwin and C. Vinoth Kumar

Department of Electronics of Communication Engineering, Sri Sivahubramaniya Nadar College of Engineering Chennai, India

E-mail: josephs@ssn.edu.in, vinothkumarc@ssn.edu.in

Abstract: Nowadays most of the information is stored in images and there is a need to convert this information into an editable format. The objective of this paper includes the development of a user-friendly tool to extract text from the video or to recognize well-written handwriting in a scanned document to an editable form. The proposed methodology is robust and can provide high-grade performance with layout distortion. The explication utilizes modalities like Optical Character Recognition (OCR), Artificial Neural Networks (ANN), Deep Learning and Maximally Stable Extremal Regions (MSER) techniques for text detection. The video extraction can be attained by using video to frame conversion and then to use the Structural Similarity Index Measure (SSIM). The prototype is developed using MATLAB and provided with a GUI, which is deployable on a workstation. The generated GUI is employed to define Region of Interest (ROI), specify the required text layout, highlighting the specific portions of the image, select the appropriate language and also export the text to a word document or notepad applications where it can be edited.

Keywords: Artificial Neural Networks, Deep Learning, Maximally Stable Extremal Regions, Optical Character Recognition, Structural Similarity Index Measure

1. INTRODUCTION

The textual data available in the images can’t be modified; the existing OCR technology allows only extracting text from a plain background (without any watermarks or distortion). The picture usually has non-textual information, and it becomes difficult for the machine to read the text frames. Most of the time, the camera may not have a front-parallel view of the scene to be captured. These images may suffer from layout and perspective distortion. Automatic text extraction is remarkably challenging as text features undergo extreme changes in size, style, orientation, and alignment. Also, the images may sometimes suffer from low resolution, layout distortion, etc. Even if the text region can be delineated well, the performance of the recognition system degrades quickly with building distortion. Thus, there is a need for a method to extract text from all backgrounds. In addition to the extraction of images / logos in the document and preserve the format to the extent possible. On the profound survey, multiple languages cannot be recognized by the same OCR engine simultaneously. The solution is proposed, a GUI created using MATLAB and also a standalone application which makes it accessible to persons unacquainted with MATLAB. It can be used in hospitals, corporate, government, educational institutions, and other businesses.
The information in the images is of very high value. The multimedia videos may contain e-learning contents, then the video from surveillance systems may have number plate information. These information extracted may provide useful to a large number of applications. The paper thus provides a solution for extricating these text in the videos into an editable format. Traditionally, all the documents have been handwritten. A method to convert these text into an editable content proves necessary since storing of all the papers confirm to be very tedious. Storing and processing of these documents manually cannot be done efficiently as by the machine. The proposed system further converts the handwritten text into an editable format from which any operation can be performed at a faster rate.

![Figure 1. Working of OCR](image)

The remaining part of the paper contains 6 sections. The second section of the paper describes the literature work, the third section explains the proposed work. The fourth section illustrates the results of the work done and the next section gives the conclusion. References are given in the section following the conclusion.

## 2. LITERATURE SURVEY

There have been many object extraction techniques but most of them are interactive methods. A user drawn line for foreground and background separation is one such method which is known as Lazy Snapping [1]. This is followed by interactive boundary refinement process. The color distribution approximation was proposed using Graph-cut method [2] based on Gaussian mixture models. User-specified bounding box are used in Grab-cut [3] to recognize object, then uses Gaussian mixture models for approximation of foreground and background distribution, succeeded by iterative energy minimization for object segmentation. But these are user interactive and cannot be applied to the proposed method. There are also segmentation methods like multi-scale convolutional network [4], de-convolutional network [5], and Mask Recurrent Convolutional Neural Network (R-CNN) method [6]. However object classification does not prove relevant to the work. Object Extraction can also be performed by finding bounding boxes and then uses multi pass method on saliency detection and foreground pixel classification to segment the objects in the boxes [7].

The information of stroke width and edges has been used most commonly for extracting text from images. One of such method is to do classification using sliding window and detect characters using Histogram of Oriented Gradients (HOG) features. Pictorial Structures Framework is used for text detection. This complete solution was proposed as an end-to-end recognition system [8]. Character candidates can be grouped together by meanings of the text line constrain [9].

Handwritten text recognition has been studied through the years using a wide variety of techniques such as stroke width, angles etc. The image captured is first segmented into words and letters and is then processed using various manner for recognition. The method used for segmentation also plays a major role since many methods are proposed for segmentation and the accuracy of the system developed on the segmentation method used. One method of handwritten text recognition was achieved using horizontal and vertical projection method and then SVM classifier for recognition of the characters [10].

Another method used is to segment text-line which is based on information energy. This energy is calculated
for every pixel in the image and then Artificial Neural Network is used for recognition. The recognition has an accuracy of 92% [11]. Segmentation can also be attained by using novel connectivity strength function, from which the text-line components are found. This system showed an accuracy of 97.30% [12]. Some of the other techniques involve recognizing handwritten text at the time of writing. But the hardware has to be redesigned for this purpose. A digital pen along with an accelerometer was designed and an algorithm is used for English letter detection [13]. The algorithm used for recognition of text has been a topic of research since many years. A Gyro-Pen [14] comprising of various sensors used in smartphones is built-in such that the path/trace of the motion of the interacting module can be traced.

Text from videos can be extracted by using mainly two methods template matching, graph matching [15]. Tracking by detection is to track the detected results in successive frames to form trajectory of the object [16] [17] [18]. Another method is to use corner response feature map to detect text regions and then use a novel fuzzy c-means clustering-based separation algorithm to separate text from complex background [19].

3. PROPOSED WORK

The development process of the proposed solution involves MATLAB based implementation, MATLAB standalone application, improving the robustness by way of Layout correction and multi-language implementation and Cloud server implementation. A user-friendly UI has been developed to ease common man usage. Fig 1 shows what the OCR actually does.

3.1. System Flow

The prototype developed is to convert a scanned document to an editable format, primarily for the English language. The tool uses MATLAB programming. Also, MATLAB based standalone application has developed. The disadvantage of existing OCR technology is that they have limited functionality and also cannot extract text composed of indigenous languages. Besides issues like layout distortion, extracting table information and multicolumn formats are not provided with proper solutions. The proposed solution yields the best solution for English and Hindi language and satisfactory results for Tamil.

The solution also ensures that robust performance is achieved for indigenous texts and also for layout distorted images. Besides this, the original format of the scanned document containing multi column text, logos, and a signature also be retained. The complete solution is developed with a user-friendly interface that can be used even by a layman and conversion into an editable format takes place at the back end. In addition to this, security log and exclusive access control is provided which forms the layers of security and prevents unauthorized access to the cloud server. Thus the entire module proposed shows better performance than existing applications at the global market and yet be easily operated by general public.

3.2. Object Recognition and Security

The method used for training the system for object detection is shown in Fig 2. The input textual image may consist of many objects such as seals, signatures, logos, etc. These can be extracted by transforming the image and then obtaining bounding boxes for the objects. Geometric transform which combines the process of translation, scaling and rotation are being used in this method. The connected components in the image are found, and then all the objects which have less than 3500 pixels are removed. Bounding boxes are then obtained for the remaining objects. The system produces a large number of bounding boxes as proposals where the system thinks that there is an object. Several boxes are created for the same object. Hence, non-maximal suppression is used to reduce the number of bounding boxes and these are extracted into a word document. The main issue here is to maintain the formatting of the text when objects are placed in between the texts. This is can be achieved by finding the words before and after the object and then inserting the object in between the words. The extracted text and the objects are then exported into a word document or notepad. Since the seals and signatures can be extracted easily, these have a higher risk of illegal practice. Two levels of authentication have been programmed to fortify the software. The password protected ingress forms the first level whereas another authentication is required at the time of exporting the result.
3.3. Deep learning
Hindi and Tamil languages are recognized by training the system character by character. This requires a very large database and the time taken may be very long if the system learns one image at a time. Hence the technique explained in [20] has been used to train the system character by character using rotation, illumination and other related methods thereby to expand the system. The system will not be able to recognize any of the character other than English characters. The system is trained by people for each character using voice inputs or other peripherals. When the system encounters an unrecognized character, the system just has to be told to which class the detected character belongs and then the system updates its database. Thus each character has to be trained extensively. The accuracy depends on the amount of training provided to the system.

3.4. Optical Character Recognition
OCR is a technology that is used to recognize text inside images and convert those texts into an editable format. OCR can extract text from scanned documents as well as handwritten documents. OCR uses two techniques like pattern recognition and feature extraction for recognition of texts. Mostly used features include detection of stroke width as in [21], before pattern recognition pre-processing is carried out in order to remove the noise from the images. Pattern recognition is used to classify the data into objects and classes depending upon the key features. The system is trained with some characters and stored in a matrix form. Once the character matches with the stored characters in the matrix the text will be detected. This technique is also known as pattern matching. Feature extraction decomposes features like lines, closed loops, line direction, and line intersections and is used to recognize the texts from an image. The characters are recognized from their shape. Feature extraction is performed after the pre-processing step, only then the characters can be recognized well. The input image is initially segmented and certain pre-processing steps are taken to avoid slight distortions like removal of noise. This is then given to the OCR Engine which is extensively trained software which works well only for English language.

![Figure 2. Object Extraction](image)

![Figure 3. Block diagram for Text-line detection](image)

3.5. Denoising
The overall technique adopted for both text recognition and Object detection is shown in Fig 3. The pre-processing block denotes the denoising process. Image denoising is used to remove the additive noise
while retaining as much as possible the important signal features. Image denoising can be done using two types of filters. Linear filter and transform domain filter. Linear filters are proved to show high accuracies during additive noise. Median filter is used to remove salt and pepper noise. Similarly Gaussian and Laplacian Filters are used for Gaussian noise. The Fig 4a shows the input image which is not corrupted by any noise. A salt and pepper noise with variance 0.15 is being added and is shown in Fig 4b and then removed using a median filter. The denoised image is shown in Fig 4(c). Thus this denoising technique could enhance the performance of the text recognition system.

![Figure 4a. Noise-free Image](image1)

![Figure 4b. Image with salt and pepper noise](image2)

![Figure 4c. Denoised Image](image3)

3.6. **Video Text Extraction**

The contents in the video are up most important in certain applications such as e-learning and surveillance systems. The video text extraction is also performed using MATLAB. To start with, the video is divided into frames at the rate of 27 frames per second approximately. These frames are then reduced to remove redundant information. Some frames are initially reduced by using the correlation property. Then the number of frames is again reduced by using SSIM. Pairs of images containing SSIM more than 50% are eliminated. Only one of the two images are retained else both the images are retained. The text extraction is then performed in these images by using the same methodology for text extraction from images. Batch processing can be used here for giving a number of images as input. The extracted results are again stored in a word file where it can also be edited.

3.7. **Handwriting Recognition**

Traditionally, all the documents were hand written and a large amount of manual effort is required to store and process these documents. Editing these documents were impossible. Hence this paper proposes a
system which can convert these handwritten documents into an editable format. A photo of the document is to be taken and this is given as input to the system. The input image is first subject to pre-processing techniques such as denoising. This is then subject to segmentation where the pre-processed image is recognized for the candidate text regions and then each character is isolated. Convolutional Neural Networks is used for recognition of the isolated characters. The details of the dataset used for training the characters are as follows. Fig 5a & 5b shows the example of the dataset collected where each letter is a separate image.

3.7.1. 55 samples of each character are taken for both capital and small letters.
3.7.2. Numbers: 0-9; 55 samples for each digit.
3.7.3. Alphabets: A-Z; 55 samples for each letter a-z; 55 samples for each alphabet
3.7.4. Hence a total of 3,480 images are collected for training purpose.

The characters from the segmented image are compared with database and the output is obtained.

![Figure 5a. Example of the Dataset taken](image1)

![Figure 5b. Example of the Dataset taken](image2)

### 4. SIMULATION RESULTS

To evaluate the performance of the proposed work, experiments have been conducted on 50 different input images and calculated the precision rate and recall rate. The proposed system has provided 90% precision rate, 97% accuracy and 98% recall rate as opposed to other methods. [22] have provided 79% precision rate and 62% recall rate whereas [18] have provided 70% precision rate. The input image taken for sample is shown in Fig 6. Fig 7 shows the dialog box wherein the user can type the username and password. Fig 8 shows the developed GUI. On clicking the Recognize option the results are displayed on the right side of the GUI and is shown in Fig 9. On clicking the Export Option the results are exported into a text document as given in Fig 13. The Add ROI (Region of Interest) button allows you to select any region and the selected region alone can also be recognized. (Fig 16). Fig 10, Fig 11 and Fig 12 shows the results for text detection and extraction respectively. The results obtained for Hindi texts are shown in Fig 15. The GUI also provides facility for loading new image, zoom in option, pre-processing, selecting a language and highlighting of text. Fig 14 shows the log file.

A standalone application was also created which allowed layman who is unacquainted with MATLAB to use the facility and also to make the solution available in systems which do not have MATLAB. The application can be opened by just clicking on to the icon ‘maatrum’ shown in Fig 16a.

#### 4.1 Handwriting Text Recognition Output

The input image taken is shown in Fig 17 and this is segmented to character by character which is shown in Fig 18. This character is then recognized using CNN where each character is compared with the dataset. A kernel of order $k \times k$ is designed for the convolution operation. The result of this convolution is then given to the middle
The textural information available in the images is not in editable format. The existing technology allows text extraction only from a plain background (without any watermarks or distortion). Images have a mix of non-textural information too, and it becomes difficult for the machine to read the text characters alone. Automatic text extraction is extremely challenging since text features undergo extreme changes in size, style, orientation and alignment. Also the images may sometimes suffer from low resolution, layout distortion etc. Even if the text region can be delineated well, the performance of the recognition system degrades quickly with increasing distortion. Thus, there is a need for a system to extract text from all backgrounds in addition to extraction of images/logos in the document and preserve the format to the extent possible. On profound survey, it is observed that multiple languages cannot be recognized by the same OCR engine simultaneously. In this paper, we have provided a solution in two platforms [1]. A GUI created using MatLab and also a standalone application which makes it accessible to persons unacquainted with MatLab [2]. The second solution is an Android Application which is Python based and allows remote access. It can be used in hospitals, corporate, Government, educational institutions and other businesses.

The remaining part of the paper is divided into 5 sections. The second section explains the literature review we have done with the third section explaining the proposed work. The fourth section illustrates the results we have obtained and the fifth section gives the conclusion for our work. The future work and references are given in sections following the conclusion.

Figure 6. Input Image

Figure 7. Authenticated Entry
Figure 8. The developed GUI

Figure 9. Recognised Output

Figure 10. Object Recognition

Figure 11. Object Recognition
Figure 12. Object Extraction

Figure 13. Exported Results

Figure 14. Log File
convolution layers. The size of the pooling layer is taken as $p_x \times p_y$. 

**Figure 15.** Recognition of Hindi Text

**Figure 16.** Specifying ROI

**Figure 16a.** Standalone Application
4.2. Layout Distortion Correction
Layout distortion is the change in the angle of the text orientation due to effects in camera during capturing process. An example of a layout distorted image is given in Fig 19. The technique followed for layout correction is very similar to the one used in [23].

The steps are,
4.2.1. The angle of distortion is initially found by subjecting the image for edge detection.
4.2.2. Thus the first text line in the image is found.
4.2.3. A straight line is then drawn on the image.
4.2.4. The angle between the text line and the drawn line is found. This is taken as “a”
4.2.5. The image is shifted to “-a” angle for distortion correction.

The input image is taken with a slight tilt angle for layout distortion which is given in Fig 20. This is then corrected using the steps explained above and the result obtained after correction is given in Fig 21. This also improves the accuracy of the OCR Engine considerably since layout distorted images produced text-line recognition but the system’s performance degraded quickly with increased level of distortion.
5. CONCLUSION

The system thus developed may produce excellent results for English and Hindi text in both noise free and noisy environments. Thus this solution may reduce the time spent in manual searching and typing process if the softcopy of the document is not available. Also this solution may be used to search for images on a large document based on a content redeem basis, rather than image retrieval mechanism. Hence this may reduce spam outputs and also the processing time. The future work may include incorporating dictionary into the system for spelling and grammar corrections. The server based system may also be implemented using Artificial Intelligence for providing a smart system which can extract the meaning from a textual image. The system can also be trained further for multiple languages.

REFERENCES

[1] Li Y, Sun J and Tang C K 2004 ACM Trans. Graph 23 303–308
[2] Boykov Y Y and Jolly M P 2001 Proc. 2001 IEEE Int. Conf. on Computer Vision 105–116
[3] Rother C, Kolmogorov V and Blake A 2004 ACM Trans. Graph 23 309–314
[4] Farabet C, Couprie C, Najman L and LeCun Y 2013 IEEE Transactions on Pattern Analysis and Machine Intelligence 35 1915–1929 URL 10.1109/tpami.2012.231;https://dx.doi.org/10.1109/tpami.2012.231
[5] Noh H, Hong S and Han B 2015 Proc. 2015 IEEE Int. Conf. on Computer Vision 1520–1528
[6] He K, Gkioxari G and Dollar P 2017 Proc. 2017 IEEE Int. Conf. on Computer Vision 2980–2988
[7] Wu K and Yu Y 2018 IET Image Processing 12 1131–1141 URL 10.1049/iet-ipr.2017.1144;https://dx.doi.org/10.1049/iet-ipr.2017.1144
[8] Wang K, Babenko B and Belongie S 2011
[9] Neumann L and Matas J 2012 Real-time scene text localization and recognition Proc. IEEE Conf. CVPR pp 3538–3545
[10] Dhande P and Kharat R 2017 International Conference on Trends in Electronics and Informatics 199–203
[11] Keisham K and Dixit S 2016 Recognition of Handwritten English Text Using Energy Minimisation", Information Systems Design and Intelligent Applications Advances in Intelligent Systems and Computing (Springer)
[12] Panwar S and Nain N 2014 IETE Journal of Research 60 432–439 URL 10.1080/
[13] Nishith A, Kotak A K and Roy 2017 An Accelerometer Based Handwriting Recognition of English Alphabets Using Basic Strokes Proc. of the 2017 IEEE Region 10 Conference (TENCON) pp 111–116

[14] Deselaers T, Keysers D, Hosang J and Rowley H A 2015 IEEE Transactions on Human-Machine Systems 45 263–271 URL 10.1109/thms.2014.2365723;https://dx.doi.org/10.1109/thms.2014.2365723

[15] Pei W Y, Yang C, Meng L Y, Hou J B, Tian S, and Yin X C 2016 Special Section On Sequential Data Modeling And Its Emerging Applications 6 19419–19426

[16] Gómez L and Karatzas D 2014 Conf. Pattern Recognit 3110–3115

[17] Yin X C, Yin X, Huang K and Hao H W 2014 IEEE Trans. Pattern Anal. Mach. Intell 36 970–983

[18] Yin X, Zuo Z, Tian S and Liu C 2016 Trans. Image Process 25 2752–2773

[19] Lu W, Sun H, Chu J, Huang X and Yu J 2018 IEEE Access 6 40198–40211 URL 10.1109/access.2018.2851942;https://dx.doi.org/10.1109/access.2018.2851942

[20] He K, Zhang X, Ren S and Sun J 2015 The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 770–778

[21] Epshtein B, Ofek E and Wexler Y 2010 Detecting Text in Natural Scenes with Stroke Width Transforms IEEE Conference on Computer Vision and Pattern Recognition pp 2963–2970

[22] Duth P S and Raj S 2018 Object Recognition in Images using Convolutional Neural Network and others pp 718–722

[23] Baumann C K R, Blackwell W B and Seales 2012 International Conference on Asian Digital Libraries 11–18