Region of Interest Extraction using Optical Character Recognition Template

Prajwal. R Athrey
Dept. of IS & E GMTI, Davanagere
Karnataka, INDIA

Asha K
Dept. of IS & E GMIT, Davanagere
Karnataka, INDIA

Rajesha. H
Dept. of IS & E, GMIT, Davanagere
Karnataka, INDIA

Sahana. S Patil
Dept. of IS & E, GMIT, Davanagere
Karnataka, INDIA

Channabasavarajan. V
Dept. of IS & E GMIT, Davanagere
Karnataka, INDIA

***Abstract*— Optical Character Recognition (OCR) is a process that converts handwritten texts into a digitally editable format that a machine can read.**

The process of OCR is discussed in this paper. We start with a scanned copy of the document; for grey scale conversion, we utilize a threshold segmentation technique, and for preprocessing, we utilize a normalization technique. The characters are then retrieved and recognized using a Convolutional Neural Network, and the retrieved data is displayed as machine editable text.

***Keywords*— OCR, HMM, BPN, CNN, Thresholding, Normalization, Segmentation.**

I. INTRODUCTION

OCR software translates a scanned document or image into a digital form of text that a computer system or machine can easily use. An OCR system is a difficult problem to solve since documents can be written in a variety of languages, styles, typefaces, and symbols, as well as contain many sorts of numbers. Converting these documents into digital form is a big difficulty because of this variety in writing. Machines, unlike our brains, which are capable of quickly detecting text, are not as intelligent as we are; making designing a system for this type of work is difficult. To address the aforementioned challenges, techniques from many disciplines of computer science, including as image processing, pattern classification and recognition, and natural language processing (NLP), are merged.

II. LITRATURE REVIEW

[1] The authors have suggested an OCR framework that is a smart phone application that has a 90% accuracy rate for handwritten documents.

[2] The authors propose a model that is built with Conda and the Tensor flow Framework. RNN has also been used by the authors to increase accuracy.

[3] Authors have broken characters down into smaller constituent graphemes using character decomposition methods. Character decomposition allows training examples to be exchanged across uncommon characters with the same graphemes, reducing the size of the Neural Network models.

III. CHARACTER RECOGNITION PHASES

To recognize text from an image, Optical Character Recognition is used. To translate a handwritten document into digital text, we can use image processing, image segmentation, image pre-processing, character extraction, and neural networks. In our work, we use OCR technology in five phases. Scanning the input text, Background and Foreground Segmentation, and Character Extraction and Recognition, a pre-processing method for raising the foreground's strength, are the processes involved.

Major Phases of OCR System:

There are five major phases, Scanning, Segmentation, preprocessing, Extraction of words and Recognition (Fig.4).

Table 1. Major Phases of OCR System

Phase	Explanation	Approaches
Scanning	Getting document image as input	Through scanner or digital camera etc.
Segmentation	Separation of input image as foreground and background	Threshold Segmentation
Pre-processing Technique	To enhance the readability of image	Normalization
Extraction of words	To extract Geometrical feature such as loops, corner points Statistical features such as moments.	Neural Network, Bayesian Network, k-Nearest Neighborhood algorithm
Recognition	To classify a character into its particular class	Convolutional Neural Network, Recurrent Neural Network

IV. METHODOLOGY

A. Scanning

A handwritten text is scanned using a digital camera or scanner in the first step of OCR. Using the thresholding process, a scanned copy of a colour image (Fig. 2) is transformed to a black and white image called a binary image. This conversion is called threshold segmentation, in which pixels with grey levels below a certain value are converted to white and those above that value are converted to black. This separation of foreground from background is then used in the next step.

$$g(x, y) = \begin{cases} 1, & \text{if } f(x, y) > T \\ 0, & \text{if } f(x, y) \leq T \end{cases}$$

- Where g(x, y) at some global threshold T is a thresholded variant of f(x, y).
- If T can change over time in the image, variable thresholding can be used.
- If T is dependent on a community, local or regional thresholding can be used
- If T is a function of (x, y), adaptive thresholding is used [5]

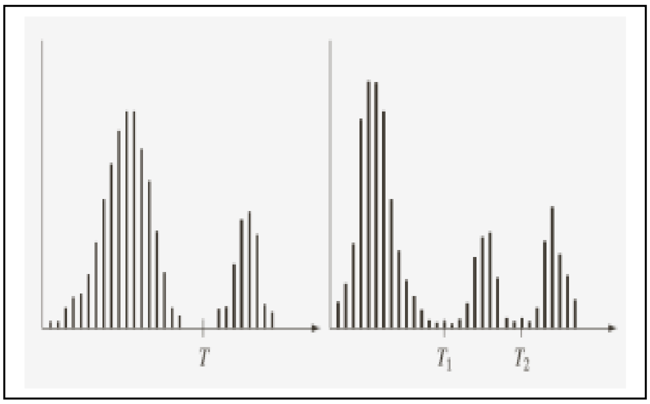


Fig. 1 Choosing the Threshold values

The image histogram's peaks and valleys will aid in determining the acceptable threshold value (s).

The suitability of the histogram for directing the selection of the threshold is influenced by a number of factors:

- the distance between peaks;
- the amount of noise in the image;
- the items' and background's relative sizes;
- the illumination's consistency;
- the reflectance's uniformity.

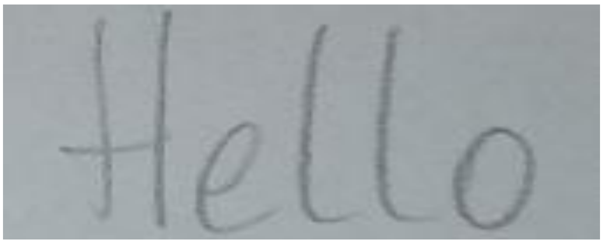


Fig.2 Scanned Image

B. Segmentation

The distinction between written text and images (Fig. 3) is made in this process. Here, all of the text created by scanning the document is segmented into components, with each word and character separated. There is noise in this scanned image. The difference in brightness or color composition in a picture is referred to as noise. It occurs when the sensor and circuitry become overheated or when the electric levels rise. The identification accuracy can be harmed as a result of the noise. Threshold Segmentation is a strategy that we used. If we want to divide the image into two regions (object and background), we define a single threshold value. This is known as the global threshold.

Fig. 3 Segmented image containing noise

We must specify multiple thresholds if we have multiple artifacts in addition to the context. The local threshold is the term used to describe all of these thresholds.

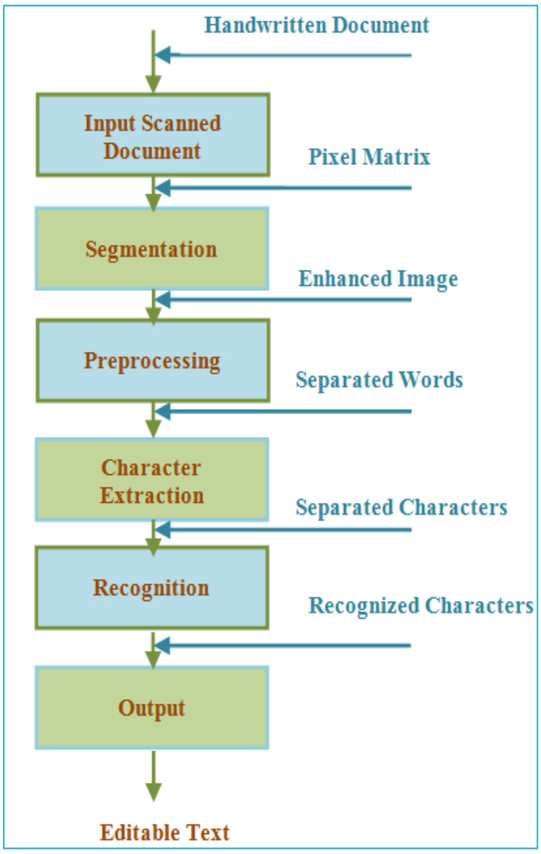


Fig. 4 Methodology

C. Pre-processing

The aim of this step is to solve the problem by pre-processing the segmented image (Fig. 5). The solution to these issues is achieved by character leveling and normalization, in which unnecessary noise is reduced using fill techniques. The term "normalization" refers to the process of adjusting the range of pixel intensity values. Matrix is used to reflect image resolution. To delete the gaps, the matrix values are modified. Pre-processing is a critical step that increases computer readability.

As compared to other approaches, Histogram Normalization (HN) performed the best

$$f(x,y)=(g(x,y)-g_{\min})\left[\frac{g_{HIR}-g_{LIR}}{g_{\max}-g_{\min}}\right]+g_{LIR}$$

.

The original image g(x, y) is extended to create the new image f(x, y). g_{HIR} and g_{LIR} are the brightness ranges of new photographs. The initial brightness levels, g_{max} and g_{min}, range from the lowest to the highest. [4]

Fig. 5 Processed images

D. Character Extraction

The fourth step is the most difficult; it involves conducting a search that allows for the detection of characters while ignoring the rest of the processed image. Many of the characters' characteristics are extracted. These characteristics are used to identify the character (Fig. 6). Geometric features such as loops, strokes, and line motions are included in this feature.

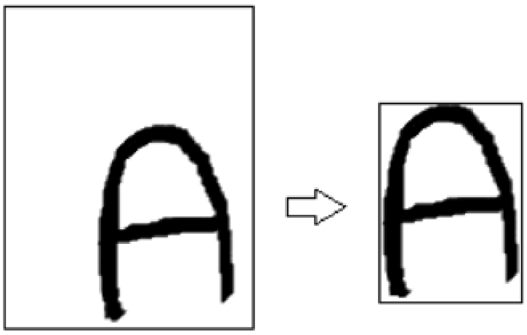


Fig.6 Character extraction

E. Recognition

The important attributes of a specific character are compared to a collection of known attributes obtained after training the OCR system with a neural network in this process, and the corresponding character is named. We get the extracted image after removing all the errors and anomalies from previous phases and joining the unconnected pixels (Fig. 7). As a result, the Optical Character Recognition process is complete.

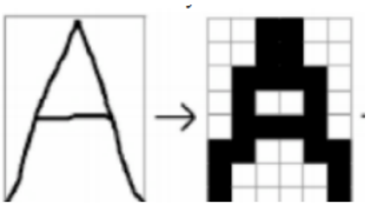


Fig. 7 Character recognition

V. CHARACTER RECOGNITION TECHNIQUES

1. HMM Approach

Hidden Markov models, or HMMs, are generative models in which the joint distribution of observations and hidden states is taken into account. The invisible procedure consists of a collection of events linked together by probabilistic evolutions, while the observable procedure consists of a set of outputs that each state can generate according to some output probability density function (PDF).

2. Neural network approach

In an optical recognition device, character recognition is crucial. To optically recognize the image and extract the characters from it, we use a Back Propagation Neural Network (Fig. 8). The BP algorithm's basic concept is that the learning process is split into two phases:

Phase I: Forward propagation

Phase II: Back propagation

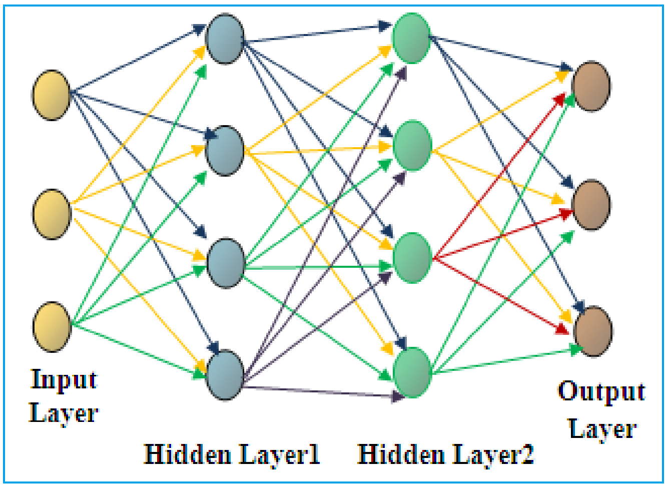


Fig.8 Character extraction using Neural Network

Text identification using the CNN formula

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - K}{s} \right\rfloor + 1$$

n_{in} – Number of inputs
n_{out} – Number of outputs
K – Convolution kernel size
p – Convolution padding size
S – Convolution stride size

3. Character Normalization

It is critical to reduce the document's character, text, and number sizes to a predetermined size (Fig. 9). We may boost the recognition accuracy of the OCR system by normalizing characters of different sizes into a fixed predefined size

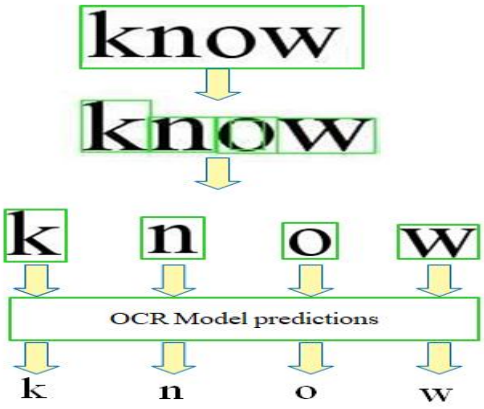


Fig. 9 Normalization

4. Tesseract

Tesseract is an open source optical character recognition engine that runs on a variety of operating systems. Between 1984 and 1994, Ray Smith and "Hewlett Packard" (HP) created it. It's written in C and C++ and works on Linux, Windows, and MacOS. HP and the University of Nevada Las Vegas (UNLV) released it as open source software in 2005.

Tesseract (Fig. 10) works in two steps. The input image is transformed to a binary image in the first step. The grey scale image made up of pixel values is known as a binary image. These pixel values have been organized in a matrix. The images are converted to a black-and-white print.

Following that, the converted black and white image is inspected, and an outline of the image is created, which includes features of the image such as image field, character location, and word length, and so on are extracted. Blobs are another name for these functions. Blobs are black-and-white image components that store the outlines of their outlines. Text detection is aided by blobs.

The text lines are divided into sections as words in the second process, and these words are then divided into characters depending on the space between them. The second phase is split into two parts: first, the system attempts to detect each word, and then these words are saved as a guide for future document detection. Final recognition attempts are made in the second step, bringing the method of identifying characters that can be used as a digital form of the text to a close.

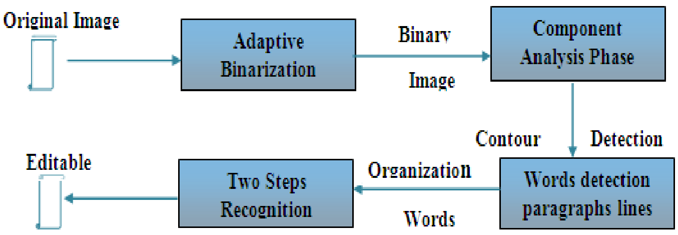
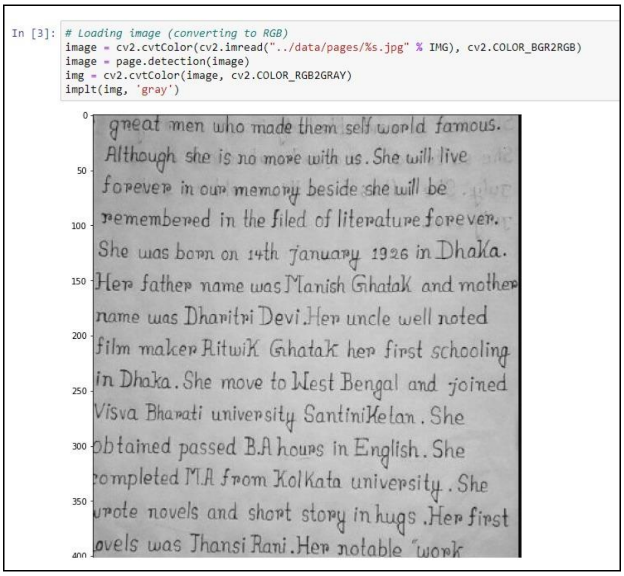


Fig. 10 Tesseract

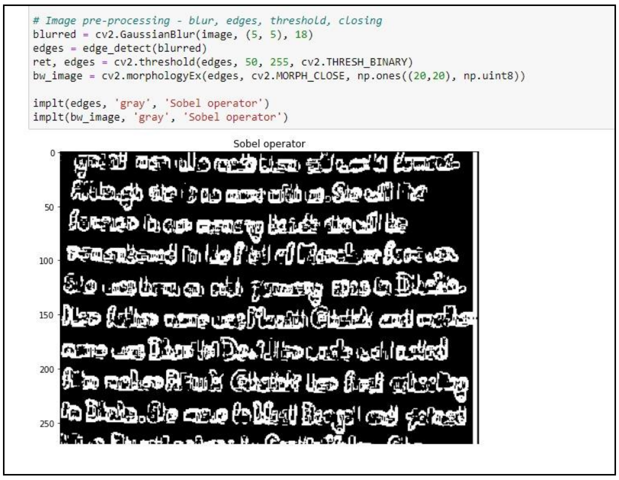
VI. RESULTS



```
In [3]: # Loading image (converting to RGB)
        image = cv2.cvtColor(cv2.imread("../data/pages/%s.jpg" % IMG), cv2.COLOR_BGR2RGB)
        image = page.detection(image)
        img = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
        implt(img, 'gray')
```

great men who made them self world famous.
Although she is no more with us. She will live
forever in our memory beside she will be
remembered in the filed of literature forever.
She was born on 14th January 1926 in Dhaka.
Her father name was Manish Ghatak and mother
name was Dharitri Devi. Her uncle well noted
film maker Ritwik Ghatak her first schooling
in Dhaka. She move to West Bengal and joined
Visva Bharati university Santiniketan. She
obtained passed B.A hours in English. She
completed M.A from Kolkata university. She
wrote novels and short story in hugs. Her first
ovels was Jhansi Rani. Her notable "work

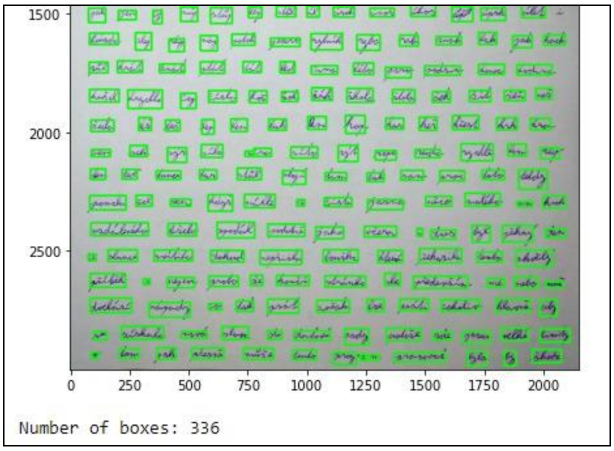
Fig. 11 Scanned copy of the Image



```
# Image pre-processing - blur, edges, threshold, closing
blurred = cv2.GaussianBlur(image, (5, 5), 18)
edges = edge_detect(blurred)
ret, edges = cv2.threshold(edges, 50, 255, cv2.THRESH_BINARY)
bw_image = cv2.morphologyEx(edges, cv2.MORPH_CLOSE, np.ones((20,20), np.uint8))

implt(edges, 'gray', 'Sobel operator')
implt(bw_image, 'gray', 'Sobel operator')
```

Fig. 12 Binary Inverted Image



```
Number of boxes: 336
```

Fig. 13 Extracting Words.

VII. CONCLUSION

In this article, we attempt to provide a brief overview of various OCR techniques. Segmenting, pre-processing, character extraction, and identification are all stages of an OCR. The OCR framework can also be used in a variety of real-time applications, such as number plate recognition, smart libraries, hospital records, and many others. Finally, the use of OCR systems in practical applications is still a hot topic of study, and it will continue to grow as newer technologies emerge.

REFERENCES

[1] Vaibhav. V. Mainkar, Ms. Jyoti A. Katkar, Mr. Ajinkya B. Upade, Ms. Poonam R. Pednekar "H ndwritten Character Recognition to obtain Editable Text", Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020)

[2] R.Parthiban,R.Ezhilarasi,D.Saravanan "Optical Character Recognition for English Handwritten Text Using Recurrent Neural Network" 2020 International Conference on System, Computation, Automation and Networking (ICSCAN)

[3] Chun Chieh Chang, Ashish Arora, Leibny Paola Garcia Perera, David Etter, Daniel Povey, Sanjeev Khudanpur "Optical Character Recognition with Chinese and Korean Character Decomposition", 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)

[4] Iza Sazanita Isa, Siti Noraini Sulaiman, Muzaimi Mustapha, Sailudin Darus "Evaluating Denoising Performances of Fundamental Filters for T2- Weighted MRI Images" 19th International Conference on Knowledge Based and Intelligent Information and Engineering Systems

[5] http://homes.di.unimi.it/ferrari/ImgProc2011_12/EI2011_12_16_segmentation_double.pdf