Int. J. Adv. Res. Sci. Technol. Volume 12, Issue 10, 2023, pp 1137-1142.

IJARST

International Journal of Advanced Research in Science and Technology

journal homepage: www.ijarst.com

ISSN 2319 – 1783 (Print)

ISSN 2320 – 1126 (Online)

Text Recognition of Bangla and English Scripts in Natural Scene Images

Mithun Dutta[*1], Akash Mohajon[2], Shaikat Dev[3], Dhiman sarker Bappi[4] and Dr. Jugal Krishna Das[5]*
Dept. of Computer Science and Engineering, Rangamati Science and Technology University, Rangamati, Bangladesh.[1,2,3,4]
Dept. of Computer Science and Engineering, Jahangirnagar University[5]*, Savar, Dhaka-1342.*

[*1]***mithundutta92@gmail.com***

ARTICLEINFO

Article history:

Received	16 May 2023
Accepted	29 Sep 2023
Available online	15 Oct 2023

Keywords:
Text Recognition,
Framework,
Localization,
Extraction.

ABSTRACT

Digital cameras on mobile devices have created several new computational hurdles. Text extraction from natural scene photos collected by such devices is an issue. This paper presents a method for recognizing English and Bangla text in scene photos. With this framework, it will be possible to recognize and locate text in its natural setting. Text recognition in natural settings is performed on multiple distinct tiers. To begin, a picture is taken using a smart device. Then text localization and text extraction are done. In the final stage, an artificial neural network is used to recognize the text from the image. This is a bilingual project, with results available in both Bengali and English. The system outperforms conventional recognition methods, producing image results with an average accuracy of 85%.

Introduction:

Scene text recognition is a complex yet hugely beneficial task that involves recognizing text in natural images. Reading words in digital photos is a difficult challenge. A challenging visual recognition issue is the recognition of characters and text in scene images. The recruitment of labels and other textual clues is a common practice in today's life and text is one of the main resources for keeping and transmitting the information. According to this, scene text recognition is not only essential for numerous information retrieval applications but also essential for human-machine communication. There are various beneficial uses for a system that can identify and detect text in real-world pictures.

Text may be bilingual and use a variety of fonts. A font may also be used in different sizes. The text could be handwritten, machine printed, or both. Our text recognition method takes an image and extracts the text zone from it, then identifies each letter and symbol in the retrieved text zone.

In this paper, we propose an approach to solving the text recognition issue that is based on recent developments in machine learning, more specifically, unsupervised computer vision algorithms.

The following paper starts with an overview of the relevant studies, describes the end-to-end text recognition method, and ends with a thorough analysis of the proposed method. Our model may be trained directly from sequence labels without the need for extensive. On image texts, it outperforms or is extremely competitive with preceding art.

The problem of text recognition has been the subject of a large number of works in recent years. Although these methods have produced encouraging results, most of them are only able to handle standard texts, which are frequently frontal, horizontal, and firmly bound. The majority of the solutions currently in use, however, cannot be widely used in practice due to the fact that many scene texts are arranged erratically in real-world applications. Here, we create a novel technique for reliably identifying either regular or irregular real texts. The following are the paper's main contributions:

1). We suggest a technique for extracting text features and character positioning cues in any direction.

2). For character recognition, we incorporate a model and a dataset. Without the need for character-level bounding box labels, the entire network can be trained directly from beginning to end.

3). We demonstrate that our text recognition-based systems are better than those that use traditional OCR.

The structure of this paper is as follows. First, we will review some related research in text detection as well as the machine learning and purpose findings that guide our fundamental strategy. Next, we explain the System architecture we established for our studies. Finally, we provide the findings of our research and draw some implications.

Literature review:

The fields of computer vision, pattern recognition, and even document analysis have made text identification and recognition in natural images important study issues in

recent years. The extraction of written text from natural photographs has been approached from a variety of unique angles by researchers from these areas.

Many different fields of study have shown a lot of interest in scene text recognition. Although exceptionally high performance on tasks like character recognition in controlled environments is already achievable, the challenge of identifying and classifying characters in complex contexts is still an ongoing research area. The majority of techniques for scene text identification and character recognition, however, rely on deftly designed algorithms unique to the new job. Researchers have put forth a variety of techniques over the past 20 years for finding messages in natural photographs. Solutions have included anything from straightforward off-the-shelf classifiers trained on hand-coded features to multi-stage pipelines [1] integrating several different methods for text identification. Edge characteristics, texture descriptors, and form contexts are examples of common features [2]. In the meanwhile, several other types of probabilistic models have been used to include many different types of prior information in the detection and identification system [3] [4] [5]. On the other hand, a few systems with extremely adaptable learning strategies try to learn all the information required from labeled data with the least amount of past knowledge. Multi-layered neural network designs, for instance, have been used for text detection and are comparable with other top techniques [6]. This is consistent with how well such methods work in older documents and handwritten text recognition systems [7].

Convolutional neural networks are indeed similar to the methodology utilized in our system. The main distinction is that this training approach is unsupervised and makes use of a considerably more scalable training algorithm that can quickly learn a large number of characteristics.

We also presented a model synthetic word dataset that is orders of magnitude different than any previously disclosed dataset. We have discussed our strategy for solving the recognition challenge in this paper. The paper represents an innovative new approach that ensures an easy recognition pipeline. Our method, in contrast to the majority of other approaches, is able of recognizing both irregular and regular texts from photographs.

System Overview

Text recognition in natural settings is performed on three distinct tiers. To begin, a picture is taken using a smart device. Then text localization is done. The next step is text extraction. In the final stage, an artificial neural network is used to recognize the text.

1. Text localization

The task of text recognition is classifying every word in the text. Finding texts inside a picture is the job of text localization. Text localization aims to properly localize text components and organize them into potential text areas with as minimal background as possible. We are using Connected Component Analysis (CCA) and typical feature types (e.g., color, edges, strokes, and texture) for text localization.

The three steps of our method's approach for localizing scene texts are as follows:

1). During pre-processing, create a text area detector to create a text confidence map from which local banalization may be used to partition text components.

2). Component analysis is presented as a CRF model in CC analysis, and the component labeling issue is then handled using a minimal classification error learning method and a graph cut inference technique.

3). Text line grouping, where inter-line edges are removed using an energy minimization model and a learning distance metric is used to build the component minimum spanning tree.

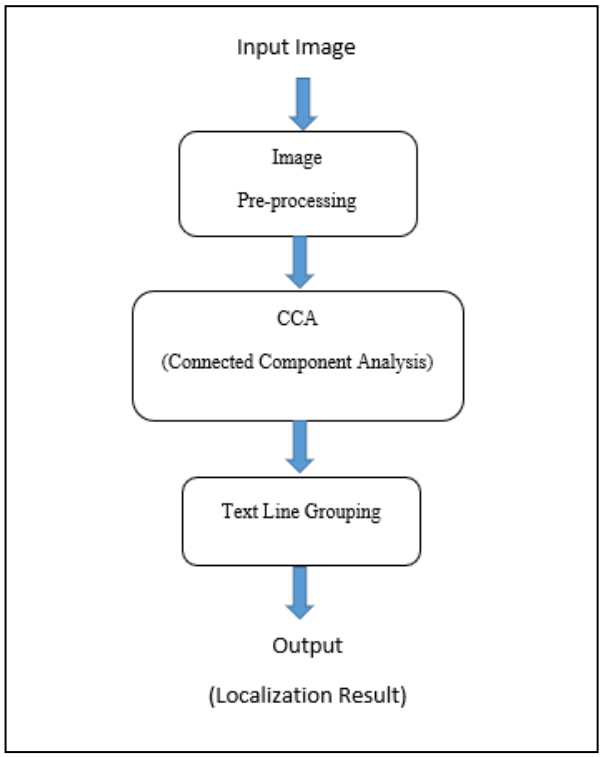

Fig 1_Steps of text localization phase

2. Text Extraction

In addition to text, a page of a document may also include graphics, tables, and other objects. It has been extensively investigated and is currently a topic of an ongoing study to extract text zones from documents. The preprocessing stage that removes uniform text areas from the paper picture is assumed in this paper, nevertheless. Every uniform text zone is divided by our system's text line and word segments. Characters and symbols are further divided into categories for words. Characters and symbols are referred to as recognition units or recognition units.

Segmentation of lines

There is white space between each text line and the text line before and after it. There are times when the top modifiers of one or more lines of text cross over with the bottom modifiers of the line before. As a result, text lines are no longer separated from one another by white space.

To divide a uniform text into its individual text lines, we used a two-pass technique. Lines are divided in the first step on the assumption that no line combination or breaking has occurred. The document's horizontal

histograms served as the foundation for this segmentation. The borders of the lines are thought to represent the horizontal gaps. All text lines generated in the first pass have their heights separated into three groups. MaxLineHt is the name for a segmented line's maximum height. These lines, which height is 80% more than the MaxLineHt, are in the first bin. The text lines with heights of more than 64% and fewer than 80 percent of MaxLineHt are found in the second bin. The third bin is for the remaining text lines. Each bin's text lines are tallied individually. The bin representing correctly segmented text lines is the one with the most text lines in it. As a threshold height, we utilize the typical height of a line of text within this bin. The fusion of multiple text lines is thought to be the cause of a text line with a height greater than the threshold height. The second pass further segments the alleged fused lines.

Using threshold height, the second pass tries to separate a potentially fused line into its component lines. The header line, the most dominant horizontal line in the image, is located by scanning it from the top to the threshold height. The fused lines are broken in the vicinity of the threshold height using the lowest pixel strength position.

Words are separated from a text line

In our dataset, a header line connects each word's letters and symbols. Because of this, word boundaries are seldom obvious. However, if any of the following characters appear in the word, there is a gap in the header line: ঞ, শ, ণ, প, গ, খ, থ, ধ, ও. Examples of such words:শেষ, গাধা, খাদ্য, ধন্যবাদ, ধর্ম. Due to the fact that the header line gap does not cause a vertical space in the word, it poses no difficulty in identifying word boundaries. Every gap of two pixels or more in a vertical histogram of a line of text is regarded as the word delimiter. It's almost self-explanatory how to divide a text line into words.

Symbolization and Characterization of a Word

Because each word has a header line, it is simple to spot the word boundaries. However, the header line unites the characters of a word, making it significantly more difficult to separate a word into its individual letters. Upper modifier symbols are located above the header line. Core letters and lower modifying symbols are located below the header line. Header line identification is necessary before segmentation can move forward. Since it dominates all other horizontal lines in a word, the header line can be easily distinguished. The top modifiers are separated from their neighbors by a vertical gap following the removal of the header line. There is a vertical gap between the characters below the header line and their neighbors. However, a character generated by this straightforward segmentation could have a lesser modifier or might be a conjunct. The phrase "preliminary segmentation" refers to this segmentation.

The conjuncts and composites characters are further divided using a "composite segmentation module.". Utilizing the script's structural components allows for the segmentation of composite characters into their component characters and symbols. An early segmentation procedure character is presumed to be a composite character based on the character's length and width in relation to other characters. If the recognition process is unable to classify an alleged composite character into established classes, further segmentation of the characters is attempted. During the first pass, words are separated into composite characters and easily distinguishable characters. Based on statistical information on the height and width of each separated box, character boxes are believed to be composite. The second pass further segments the hypothetical composite characters.

Preliminary word segmentation

A word is made up of all its letters and symbols together. Therefore, the header line must be "removed" before segmentation is attempted. Two issues arise if the header line is deleted from the word in a single motion. First, certain character pairings simply vary in the heading line region. One person's header line is incomplete, while another person is complete. It is challenging to tell one from the other after eliminating the header line. Second, a slight tilt in a word's orientation, if it exists, might make the header line wider. The top few characters are cut off if the header line is totally gone. On the other contrary, if just a substantial portion of the header is deleted, the remaining header line becomes background noise for the digits.

A word's horizontal histogram is used to locate the header line to solve both of these issues. But each character is stripped of its own header line. When searching for a vertical gap, the area above and below the header line is disregarded in to obtain the individual characters. It is handled differently for the picture below and above the header line. An image's vertical histogram is created for each component. A vertical gap between two neighbors is represented in the vertical histogram by a column that has no black pixels in it.

Temporary Character Information: Height and Width

To identify the character boxes that are most likely to be composite character boxes, preliminary segmentation statistics on the width and height of characters are employed. Based on their breadth, all characters are classified into three groups. The term "MaxCharHt" refers to a segmented character box's maximum height. Character boxes that are at least 80% of MaxCharHt tall are placed in the first bin. Character boxes that fall between 80% and 64% of MaxCharHt in height are placed in the second bin. The third bin should contain the remaining character boxes. Each bin's character boxes are counted individually. The representative bin for correctly segmented character boxes is the bin that has the most character boxes in it. The threshold character height category is set by the character box's average height in this bin. All characters are identified as having potential lower modifiers if their height exceeds the threshold character height. The threshold character width is calculated in the same way. Any character that is wider than the threshold character width is thought to be a shadow character or a conjunct.

3. Text Recognition

The text recognition model is comprised of convolutional layers, GRU layers, and dense layers. The purpose of

using the convolutional layer is twofold: first, it learns to extract better features from the input data, and second, it reduces the time dimension of the data. In theory, a convolutional layer should produce more robust features, causing the subsequent layers to produce better predictions, and it reduces the time steps of the data, allowing the subsequent layers to do less work because there are fewer time steps, resulting in a faster network overall. Training requires a very long time when the depth of neural networks increases, and the models experience severe optimization problems. To overcome this problem, we used batch normalization as a way to speed up training and enhance generalization. After each convolution layer, we added a batch normalization layer, and after the batch normalization layer, ReLU activation is used to pass data to the next layer. After convolution layers, there are GRU layers used in this model. Hyperbolic tangent function is used as the activation function for the GRU layer. After GRU layers, dropout is used to avoid a possible overfitting issue. The probability of a layer's outputs being dropped is set to 0.5. After the GRU layer, two dense layers are used. The ReLU activation function is used for the first dense layer. After the first dense layer, a dropout of 0.5 is applied. The last dense layer is utilized to combine the local knowledge gained by the model to accomplish class recognition. So, a dense layer with SoftMax activation is implemented to work as the model's final or output layer. So, the last dense layer with a SoftMax activation function produces a likelihood for every one of the characters that the model is attempting to predict. The output layer outputs a probability of 65 characters (vowels, vowel symbols, consonants, consonant symbols, spaces, etc.). The model is character-based, meaning it will output characters instead of word probabilities. Outputting character probabilities is more efficient because we only have to worry about 65 probabilities for each output instead of, say, a hundred or thousands of possible words. Following the dense layer, a CTC decoder with the greedy search method is used to decode output sequences.

Results and Discussions

Text Detection: Using MSVC++ and the OpenCV computer vision library, we accomplished the recommended idea in the Windows platform. We are creating a standard database of outdoor scene photographs with texts in English and Bangla because there isn't any available database. These images are taken with a digital camera and a smart phone. These photos are taken from different scenarios like roadways, institutions, wall writings, vehicle stations and clothes paintings, etc. The focus of these photos is kept on the text, which is intentional. These include English and Bengali text in a variety of typefaces, sizes and orientations. To evaluate the results of the suggested approach, we used 15000 sample photos collected by us. Ten thousand photos make up the training set and the remaining 5000 images make up the test set for the experiment. Beyond the text in Bangla, several of these pictures also include English text. Fig. 2 displays the detected texts in a few of these samples.

Fig 2_Captured image and text detected image

The proposed method also has the benefit of identifying word boundaries, which should aid the later steps of the proposed system.

Text Extraction: The resultant images of previous step (text localization phase) are used in this step. The photos in Fig. 3 are just a few of those on which the algorithm successfully extracts all of the Bengali and English text components. There are more than twelve thousand of these images, all of which have relevant text elements that can be retrieved. In contrast, less than 3000 of the experimental photos on which our algorithm performs quite poorly are.

Fig 3_Text detected image and text extracted image

Text Recognition: Images from the previous step (Text Extraction step) were used here. These photographs were utilized for both training and testing. The proposed neural net has been used for character recognition. Fig.4 shows some images of the overall process (from capturing to recognition).

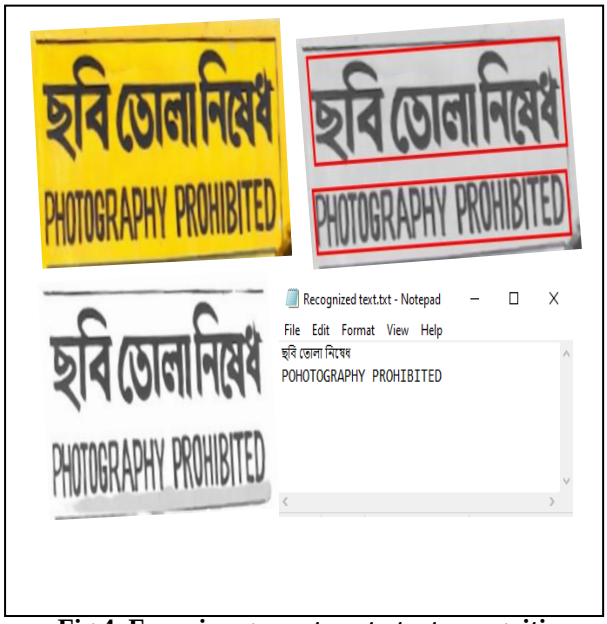

Fig 4_From image capture to text recognition

Table 1 summarizes the suggested system's accuracy in terms of its steps.

Steps	Text Detection	Text Extraction	Text Recognition
Total Samples	15000	15000	15000
Correct for Bengali	12430	12010	11970
Accuracy for Bengali	82.87%	80.07%	79.80%
Correct for English	13110	13030	12710
Accuracy for English	87.40%	86.87%	84.73%

Table 1. Proposed system's accuracy in various steps

The system recognizes both English and Bengali languages, and it may be manually tweaked to identify either English or Bengali. The number of successfully recognized photographs was used to calculate the sample size and accuracy. Figure 5 depicts the correct results of all steps together.

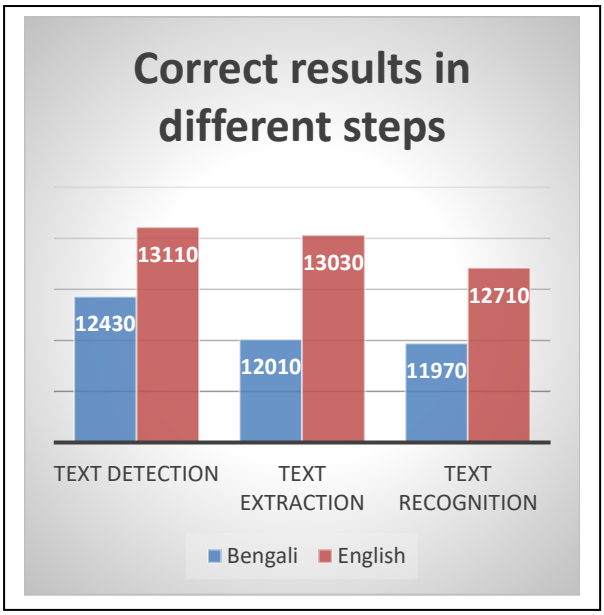

Fig 5_Correct results in different steps

Conclusion

A comprehensive system for Bangla and English text recognition from images is presented in this paper. Prior to the system detecting the texts, images are initially taken from the surrounding environment. Then the detected texts are extracted from the image. Finally, text recognition was accomplished by a neural network.

Even yet, the simulation outcomes of the suggested system on our dataset of outdoor sceneries with texts in Bangla and English are promising. However, there were a number of instances where it gave false positive results or was unable to recognize some of the words or portions of them.

Again, the suggested method performs effectively even on Bangla and English text that is curved or tilted. The proposed method, however, won't work if the size of such slanted or tilted text is too small.

In the future, we plan to use a larger dataset so that the resulting system can address before mentioned problems as well as improve performance.

In the future, we plan to use a larger dataset so that the resulting system can address before mentioned problems as well as improve performance.

References:

1. X. Chen and A. Yuille, "Detecting and reading text in natural scenes," in Computer Vision and Pattern Recognition, vol. 2, 2004.

2. T. E. de Campos, B. R. Babu, and M. Varma, "Character recognition in natural images," in Proceedings of the International Conference on Computer Vision Theory and Applications, Lisbon, Portugal, February 2009.

3. J. Weinman, E. Learned-Miller, and A. R. Hanson, "Scene text recognition using similarity and a lexicon with sparse belief propagation," in Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 10, 2009.

4. J. J. Weinman, E. Learned-Miller, and A. R. Hanson, "A discriminative semi-markov model for robust scene text recognition," in Proc. IAPR International Conference on Pattern Recognition, Dec. 2008.

5. X. Fan and G. Fan, "Graphical Models for Joint Segmentation and Recognition of License Plate Characters," IEEE Signal Processing Letters, vol. 16, no. 1, 2009.

6. Z. Saidane and C. Garcia, "Automatic scene text recognition using a convolutional neural network," in Workshop on Camera-Based Document Analysis and Recognition, 2007.

7. Y. Pan, X. Hou, and C. Liu, "Text localization in natural scene images based on conditional random field," in International Conference on Document Analysis and Recognition, 2009.