The International Journal of Engineering and Science (IJES)
|| Volume || 8 || Issue || 8 Series I || Pages || PP 81-86 || 2019 ||
ISSN (e): 2319 – 1813 ISSN (p): 23-19 – 1805

Improved Template Matching Techniques for Unicode Character Recognition

Patrick Obilikwu[1], Karim Usman[2] and GbengeAondoakula Raphael[3]
[1]*Department of Mathematics and Computer Science, Benue State University, Makurdi, Benue State, Nigeria*
[2]*Department of Mathematics and Computer Science, Benue State University, Makurdi, Benue State, Nigeria*
[3]*Department of Mathematics and Computer Science, Benue State University, Makurdi, Benue State, Nigeria*
Corresponding Author: Patrick Obilikwu

----------ABSTRACT----------
Image text recognition is a process of getting text values out of an image. The extracted characters save space and enable editing on the characters. This is also called optical character recognition (OCR). Optical recognition is performed off-line after the writing or printing has been completed, as opposed to on-line recognition where the computer recognizes the characters as they are drawn. Template Matching is a system prototype useful to recognize the character or alphabet by comparing two images of the alphabet. This work focuses on providing high accuracy on offline recognition character using a template matching technique which does not adapt for some writing style. It has been implemented with Java programming language and opencv library. Template matching technique is modified to adapt many writing styles with an unending template creation opportunity and still maintaining all other properties of template matching. The new template matching techniques adapts several writing styles since it allows template creation at the recognition. .The accuracy ranges zero (normal) to a hundred per cent (improved accuracy). Since multiple templates are involved in the system, the system used a run-length compression algorithm to compress its templates. The processes are starting from gathering data (image), segmenting into individual character image, comparing with the templates or creating a template image and lastly producing an editable text. The text produced gave high accuracy. The future work is to enable online data storage and configure the system for small devices.
KEYWORDs;- *OCR, template matching, compression algorithm, segmentation, improved accuracy*

Date of Submission: 05-08-2019 Date of acceptance: 20-08-2019

I. INTRODUCTION

Images that contain text cannot be easily edited, and storing them occupies more space, as compared to the edited text of the same image. The need to turn the text in images into ASCII coded characters or equivalent arise for editing and saving spaces after being scanned. This type of conversion is called offline recognition. Online recognition involves the automatic conversion of text as it is written on a special digitizer or (PDA), where a sensor picks up the pen-tip movements as well as pen-up/pen-down switching [1].Optical character recognition follows an algorithm to do the recognition, some of the algorithms for offline recognition include: template matching algorithm, structural algorithm, statistical algorithm, neural network algorithm, supported vector, decision tree classification [1, 2].A template is an image of a character or alphabet. Template matching is the use of stored array template image or image attribute to compare with the inputted image character to determine the character it represents (e.g. represent in ASCII) [1, 2]. This involves matching two images of an alphabet. The stored template can be compressed to save storage. Data compression techniques can be an either lossless or lossy category. Run-length is lossless generic data compression technique (i.e. it does not lose data on compression) [3]. This research focuses on increasing the accuracy of character recognition in template matching. The test will be done on a typed documents that are later in image format, when scan from their hard copy, and where segmentation and noise removing are easily carried out for further processing.

Template matching is an OCR technique that stores template images for its character recognition [4, 5]. The system works only on stored templates [6]. This method is not adaptive to differences in writing style and it suffers from sensitivity to noise [7]. Hence need to store several writing styles to recognize accurately. Ravindra (2011) earlier complain of large storage size required by neural network classier and used template matching with predefined template font style (that is one template for each alphabet). Since the study is storing several templates, therefore, it needs more storage.However, researchers' fails to consider the size of templates, therefore this study will address the storage size of a template. A compression algorithm will be applied to improve the template size rather than increase the hardware component. This study aims to convert an image that has text to editable ASCII text and add several templates on a character.

II. RELATED WORKS

Pugazhenthi and Vallarasi. [5] researched Printed Tamil Text using Template Matching Method of Bamini Tamil Font and used correlation coefficient for similarity measurements of the template. The research shows that it created a template image of size 42X42 to be stored for recognition.Each character was segmented was correlated with the preloaded templates of the system. The maximum correlation judges the character. In the same way, every segmented input is checked with the preloaded templates. These templates are mapped onto Tamil Unicode for recognition and further process to an editable text file. However, the system loses its accuracy when the font size in the document image is small (considerably with less than 20 pixels).

Ravindra. [4] researched printed Kannada numerals, where it contains only ten digits. A training data set for each numeral is created using Nudi 17 K font and stored in the database. The numerals in this data set have a size of 42×24. Each image template is matched with the stored numeral image data in the database. The researcher used a bounding box for each character and stored the content in the box. He conducted experimentation on 30 different fonts of Kannada numerals generated from Nudi 4.0 software. He found out that two numerals were detected with 100% accuracy. The worst detection rate is shown by a numeral with an accuracy of 76.67%. The overall accuracy of all the numerals using the research was 91%.The researcher stated that the failure of the system is due to the clarity of the character (characters that have been broken and font having sharp edges and corners). Although it did not cover much character set it recognizes the different font.

Rachit.[6] developed a prototype for the Optical Character Recognition (OCR) system and implemented the Template Matching algorithm. He used two greyscale image as input and one coloured image. The result of the output was good but had the following limitations:

i. The system prototype has some limitations related to performance.
ii. It works only with stored templates of alphabets and numbers with fixed-sized templates.
iii. It works only for greyscale image.

Rajithkumar and Mohana. [8] made OCR for stone in the script of Kannada character of Different Time Frames with template matching. They used a mobile camera of 16 Megapixel resolution camera to capture characters and cross-correlation technique was implemented in matching the characters coefficient. MATLAB (R2009a) was the software tool used for Recognition of Kannada Characters.According to the research, template matching torrent noise than a structural algorithm, and hence they used a template image of size 42 X 24. The resulting accuracy in recognizing Stone inscriptions characters of both Hoysala, Ganga time frames and with better time efficiency was about 92% per cent. The system was restricted to only two characters of Hoysala and Ganga time frames.

Azher. [9] recognized Bangla handwritten characters using a template matching algorithm and improved the accuracy by using a normalized cross-correlation (NCC). The required images are acquired using a digital camera. The normalized cross-correlation function between the images pair can be defined in the discrete case as follows.

Where t is a template image and t' is the average of the binary image. The researcher assumed that x is the

$$NCC = \frac{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(x-\bar{x})(t-\bar{t})}{\sqrt{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(x-\bar{x})^2.(t-\bar{t})^2}}$$

handwritten image, having the same size as the template and x' is the average of the binary image. All experiments were done on dual-core 3.00 GHz with 2 GB RAM under the MATLAB environment. In the experiments, 50×25=1250 images were employed and the size of the input images was 480×320 pixels. The total recognition rate was 93.92%. However, the system used images of characters with large font size (480×320 pixels) only.

Ayushi and Vinaya. [10] performed a recognition of handwritten code using Template Matching and used the correlation coefficient for similarities measurement. Database of 30 samples of handwritten text

received from four people was scanned by a scanner for template dataset of handwritten characters in various fonts and size, which includes; lower case letters, numbers and selected special characters.The research was built in MATLAB and works by taking an input image, then preprocesses the input image and recognizes the characters from the trained data to give an output of the given code as printed text. In the system, the accuracy of the project can be improved when expanded to include a wide variety of handwritings and bigger set of handwritings which enables us to encompass a large set of variations. This was a recognition of a single character in an image.

III. THE PROPOSED MODEL

The proposed system noted the adaptability problem of template matching and hence provide an easy way of adding template at run time of the system to enhance the accuracy of the character recognition and adaptability of several styles. Also, the system is mindful that storing several templates occupies much space and therefore compresses its template.

The new system is intended to be interactive such that it will ask for the user "what is the inputted character image" when its lowest accuracy level set is not reached. By this, templates will be added to the system regularly. The system will be useful in any character set, asked to recognize since recognition and template creation happens concurrently. The system stages that are used to get machine editable characters are explained in the next section.

IV. STAGES OF OPTICAL CHARACTER RECOGNITION(OCR) IN THE NEW SYSTEM

The stage used to get machine editable text in the system include data gathering, segmentation, Character recognition and template creation, and lastly the machine editable text. Figure 1 below shows how the stages are linked to another

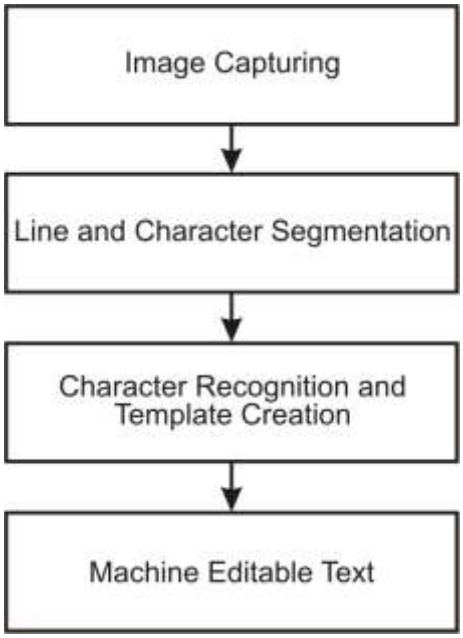

Figure 1:Stages in Optical Character Recognition

a.Image Capturing: This defines how data will be inputted to the system. The data in the system is an image that contains text. The image will be created from screenshots with a white text and a black text computer typed text. The system will receive the image and move to the next stage.

b. Line and character segmentation: Line segmentation: this is separating various lines of character in an image. A trim function is set to remove white from top and bottom of the questioned image. A function is set such that it checks a row that contains only white colour starting from the top, and when this is found it returns the row index. An OpenCV function "submat" used to crop the image from start to the returned row index. This result in two images, the cropped image (a line) and the remaining part of the image. The process is repeated on the remaining image until the remaining image is also a line. Character segmentation: A function is set to trim the given image from the left and right of the line image. A function again is set such that it checks a column that contains only white colour starting from the right, and when this is found it returns the column index. An OpenCV function "submat" used to crop the image from left to the returned row index. This result in two

images, the cropped image (a character) and the remaining part of the line image. The process is repeated on the remaining line image until the remaining line image is also a character. After a character is found a trim function is used to crop the character image from all side resulting in a design character sent to the recognition and training phase.

c.Character Recognition and Template Creation: This state iterates through all the character images from segmentation stage and processes them independently. Each character is either recognized by the new system or a template will be created. The enhanced algorithm to achieve this process is as shown below;

Enhanced Template Matching Algorithm

1. start
2. Set the accuracy level needed
3. A character image from segmented charterer images and the highest matched is set to zero.
4. Select a template from the file.
5. check difference test, if not satisfied repeat the fourth step
6. Rescale the character image and the template to the same size, the matching metric is computed and stored as recent matched
7. Then the best match found is stored as highest matched (between highest matched and recent matched).
8. If the image is not matched (highest matched not 100%) and template been available, repeat the fourth step.
9. Accuracy less than best matched?
10. Then the best match template corresponding character is stored as the recognized character and moved to step twelve.
11. Ask for character or characters that appear and store input as recognized character and store the character image as new template corresponding inputted character.
12. Stop

Figure 2. Enhanced Template Matching Flowchart

d.Producing machine editable text: This is the stage where characters are received one after another as they pass through the previous stage "character recognition and template creation". The line coordinate of the line image in the segmentation stage is used to separate line with the end line character in java ("\n") when the next character is a full stop or full colon. The word separation is done by dividing the biggest character spaces (character spaces of all lines) between characters by two. The result is now the actual character space separated with space using (" "). At this stage, every character image would have been converted to string. The recognized string will be displayed in their editable formats where minor correction can effected by the user if necessary. Thereafter, the user can save the string in any format of their choice like TXT, HTML, CSS, PDF, etc.

V. METHODOLOGY

This system is an OCR system that used template matching techniques for its character recognition. A typical OCR system consists of several components as shown in Figure 1. [7]. The first step is to digitize the analogue document using an optical scanner. When regions containing text are located each symbol is extracted through the segmentation process.

The extracted symbols are pre-processed, eliminating noise to facilitate feature extraction. The identity of each symbol is found by comparing extracted features with descriptions of symbol classes obtained through a previous learning phase. Finally, contextual information is used to reconstruct words and numbers of the original text. (Arindam et al, 2017, p. 15).

VI. RESULTS AND ANALYSES

In this section, an image page of the different font from the one in the file is loaded into the system and accuracy is set to 95%. The system automatically detects the different writing style and request input from the keyboard and several templates are added to the template resource folder.

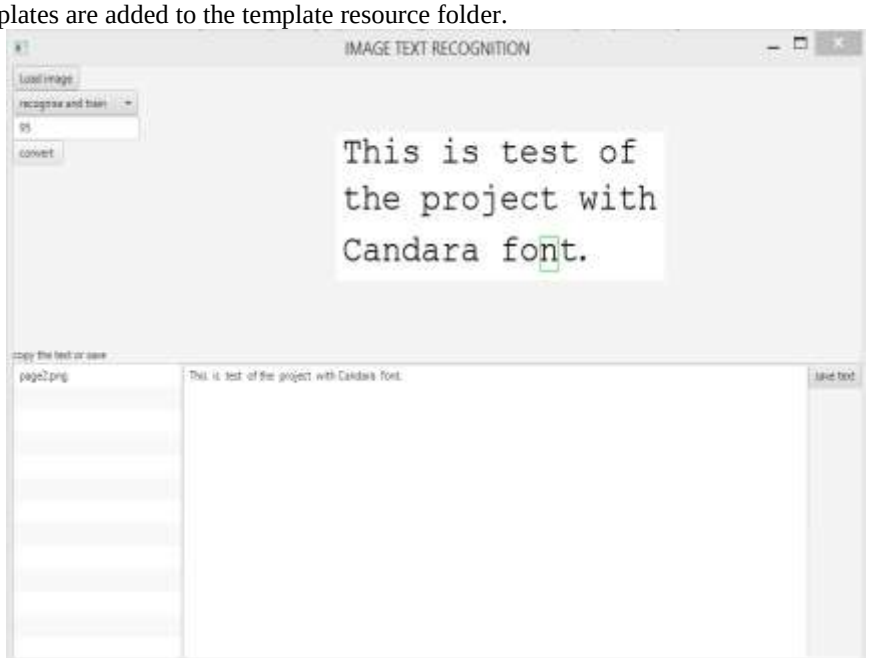

Figure 3: Conversion of the new image Page with the different writing style

VII. CONCLUSION AND RECOMMENDATION

The enhanced optical recognition of this system is done here with template matching and it yielded a great accuracy level since it allows for new templates to be added to the system at recognition time. Unlike like most systems, this system does not any previous knowledge of the character set to be recognized. Hence, it is capable of recognizing a wide spectrum of character set. The enhanced Template Matching is capable of recognizing characters up to 95% accuracy level. Advanced techniques like neural networks could be implored in this character recognition system to improve to 99% accuracy level. Again, more techniques could be implored for accurate segmentation processes.

REFERENCES

[1]. Vithlani, P. J. (2015). A study of optical character patterns identified by the different OCR algorithms and generation of a model for the elimination of deficiency in identifying the patterns of optical character. PhD theses. Saurashtra University. Retrieved from http://hdl.handle.net/10603/130552.

[2]. Arindam, C., Krupa, M., Pratixa, B. and Soumya, K. G. (2017). Optical Character Recognition Systems for Different Languages with Soft Computing. Retrieved from https://www.researchgate.net/profile/Arindam_Chaudhuri2/publication/321518201_Optical_Character_Recognition_Systems_for_Different_Languages_with_Soft_Computing
[3]. Brooshear, G. J. (2009). Computer Sciences: An Overview (10th ed) Michael H. London, Greg T
[4]. Ravindra, S. H. (2011). Template Matching Approach for Printed Kannada Numeral Recognition.
[5]. Communications in Computer and Information Science3(4), 648 -651. Retrieved from https://www.researchgate.net/profile/Ravindra_Hegadi/publication/230639908_Template_Matching_Approach_for_Printed_Kannada_Numeral_Recognition
[6]. Pugazhenthi, D. and Vallarasi, A.S. (2015). Offline Character Recognition of Printed Tamil Text using Template Matching Method of Bamini Tamil Font Indian Journal of Science and Technology 8(35). Retrieved from http://www.indjst.org/index.php/indjst/article /download/86811/66582
[7]. Rachit. V. A (2013). Optical recognition using template matching (Alphabets and Numbers). International Journal for Computer Sciences Engineering and Information Technology Research 3(4), 229 -230. Retrieved from http://www.tjpc.org/publishers/2-14-1381407981 -29.%20Optical%20character.full.pdf .
[8]. Ning LI (1991). An Implementation of OCR System Based on Skeleton Matching. Retrieved from https://kar.kent.ac.uk/21129/
[9]. Rajithkumar B. K. and Mohana H.S. (2014). Template Matching Method for Recognition of Stone In scripted Kannada Characters of Different Time Frames Based on Correlation Analysis. International Journal of Electrical and Computer Engineering (IJECE) 4(5), 719-729. Retrieved fromhttp://www.iaescore.com/journals/index.php/IJECE/article/ viewFile/5522/4839
[10]. Azher, U. (2014). Handwritten Bangla Character Recognition Using Normalized Cross Correlation, IOSR Journal of Computer Engineering (IOSR-JCE) 16(3), 55-60. Retrieved from www.iosrjournals.org
[11]. Ayushi, C. and Vinaya, S. (2016), Implementation of Handwritten Character Recognition using Template Matching. Journal of Current Engineering and Technology 6(6), 250-252. Retrieved from http://inpressco.com/category/ijcet

Patrick Obilikwu" Improved Template Matching Techniques for Unicode Character Recognition" The International Journal of Engineering and Science (IJES), 8.8 (2019): 81-86