ISSN:2229-6093

Saud M.Maghrabi , International Journal of Computer Technology & Applications,Vol 8(5),602-608

An Offline Arabic Handwritten Character Recognition System using Template Matching

Saud M. Maghrabi

Department Of Computer Science,
College of Computer and Information Systems,
Umm Al-qura University, Makkah, Saudi Arabia.
smmaghrabi@uqu.edu.sa

Abstract

Abstract : **This paper proposes computerized offline handwritten Arabic text recognition method using template matching technique based on 2-D normalized cross-correlation. The objective of this research is to find efficient and accurate handwriting Arabic text recognition algorithm, which can accept handwriting input and recognize handwritten character entered in the computer using template matching technique. The recognition process consists of five stages: input capture, image preprocessing, line segmentation, feature extraction, character recognition. These stages are implemented in MATLAB R2017a version. Experimental results demonstrate the recognition adequacy isolated test dataset with a general exactness of 97% for Arabic handwritten characters.**

Keywords : Template matching, Normalized Cross-Correlation, Arabic handwriting recognition.

1. Introduction

Optical character recognition (OCR) is very popular research field since 1950's [1]. OCR is a field of research in pattern recognition, artificial intelligence and computer vision [2]. It is a procedure of changing printed or handwritten scanned reports into ASCII characters that the computer can perceive. Character recognition is a process of converting handwritten, typewritten or printed text images into machine encoded code or text [3]. Most of the published researches on recognizing printed and handwritten text have been on English character, but Arabic character recognition has not been covered as perfectly as English. Arabic character recognition is a specialty of identifying, segmenting, and distinguishing characters from the image. It is getting increasingly consideration since a decade ago because of its extensive variety of use. It is an automated process that enhancing the interface amongst man and machine in numerous usages, such as postal code recognition, automatic data entry into large administrative systems, banking, automatic cartography and reading devices for blind [4]. Parvez M.T. et. al. [5] reported a review of the earlier methods in Arabic handwritten character recognition for the period 2005-2011. Usman Saeed [6] presented a survey of the previous works in Arabic handwritten character recognition for the period 2011-2013.

Script recognition can be characterized in two primary classifications as per the way that characters are sustained to the framework: online and offline recognition. The offline class recognizes scripts after the composition is finished whether the script is written by hand or machine, by difference, online script recognition perceives scripts while the user composes. The execution is reliant upon the nature of the information archives.

The pattern of Arabic handwriting document is changeable from one writer to another. Essential features of Arabic writing are:

- Arabic character is written from right to left, however, Arabic numbers are written from left to right.
- There are 28 Characters in the Arabic script, whose shapes relying upon their location in the word (isolated, begin, middle, and end).
- An Arabic word must be written recursively and each character must be connected to another one [7].
- A few Arabic characters have a similar shape, and are separated just as far as the number and position of dots on the characters.
- Many Arabic characters scripts are cursive and they can be overlapped.
- High level of variety in Arabic composing styles of people.

Therefore, methods for other scripts are usually unsuitable for Arabic.

This paper exhibits a technique for off-line handwritten Arabic character recognition system. The system comprises five stages: input capture, image preprocessing, line segmentation, feature extraction, character recognition. It accepts handwritten Arabic character as input, and then processes the character to

recognize the pattern and finally adapts the character to a good form of input. This work is restricted to Arabic characters. The objective of this paper is to build up a progression of many algorithms to the point of making a framework for computerized offline handwritten Arabic text recognition method using template matching technique based on 2-D normalized cross-correlation. A character is extracted from an input image and normalized. For recognition process, the extracted character is compared with all templates in the database to locate the highest similarity of the input character.

The matching is calculated using 2-D normalized cross-correlation method to classify similarity between the input image and the database images (temples). There arises a template for all conceivable input images. Experimental results illustrate that the proposed method is efficient for recognition Arabic handwritten characters.

The rest of the paper is sorted out as follows. Section 2 presents the related works; Section 3 describes the proposed method; Section 4 explains results evaluation and discussion; and section 5 concludes the paper and provides possible future works.

2. Related Works

Several works are described in the literature using template matching technique for classification or recognition of characters for various languages. In this section, some of these methods are briefly reviewed.

Majid Ziaratban et. al. [8] proposed a based template matching method for recognition of Farsi/Arabic numerals utilizing neural networks system and multi-layer perceptron show asserted an exactness of 97.65%. Nevertheless, the testing has not measured the mind boggling script for recognition. Sunny Kumar et. al [9] had studied the execution of template matching algorithm to English handwritten and type written characters utilizing parameters like exactness rate and time taken for execution. The exactness achieved is observed to be around 83% for the both, but the examination was conveyed for a little data set, i.e., on around 360 images, which is moderately less to test template matching algorithm. Nikhil et. al. [10] used the template for multi textual styles and multi font sizes of English script and achieved a precision of around 90%. Mo Wenying et. al. [11] applied the template matching calculation by customizing as for weighted matching degree. This calculation gives a higher matching rate and beat the erroneous recognition delivered by traditional technique with exactness of around 100%. Jatin et. al.[12] utilized the template matching technique for type written English characters and grouped classified neural network classifiers. Soumendu et. al. [13] had proposed a calculation for Japanese character recognition utilizing the center of gravity features and Euclidean distance features and character with least Euclidean distance is the feature employed for character recognition. Mahabubar et.al [14] had proposed a strategy for recognizing Bangla handwritten characters using the convolution neural networks. N. Shobha Rani et.al [15] proposed methodology makes an attempt to analyze the performance of the template matching with required enhancement particular to Telugu character recognition system, and the accuracy obtained with test templates is around 93.55% for the sample size of around 2730 different characters from which 2554 characters are recognized correctly. Seema Barate et.al [16] implemented pattern matching system for English language text character recognition. The system has image processing modules for converting text image to generate .doc file. It matches the character with the trained dataset using template matching algorithm .Then generate the character sentence by using template generator. Then generate document file (.doc file). Kishori Kokate et.al [17] Used English optical character recognition and pattern matching technique, specifications of the products are extracted and saved in tabular format. This system is used to find out location of the product.

Kajal Gade et.al [18] proposed matching system for Devnagari (India) language text character recognition. The technique is for OCR system for different five fonts and sizes of printed Devnagari script which will be hardware related. The recognition rate of the proposed OCR system with the document of image of Devnagari Script has been found to be quite high.

The majority of the depicted works demonstrate that the template matching strategy is executed for various language scripts and the most techniques are focused on English character recognition. Along these lines the proposed strategy creates a push to concentrate the usage of the template matching procedure to an Arabic character recognition framework. The proposed framework shows the recognition productivity on my own test dataset with a general precision of 97% for Arabic handwritten characters.

3. Methodology

In this paper, a system is proposed for Arabic handwritten character recognition. The stages of the proposed system are implemented using Matlab R2017a version according to the structure chart shown in fig 1.

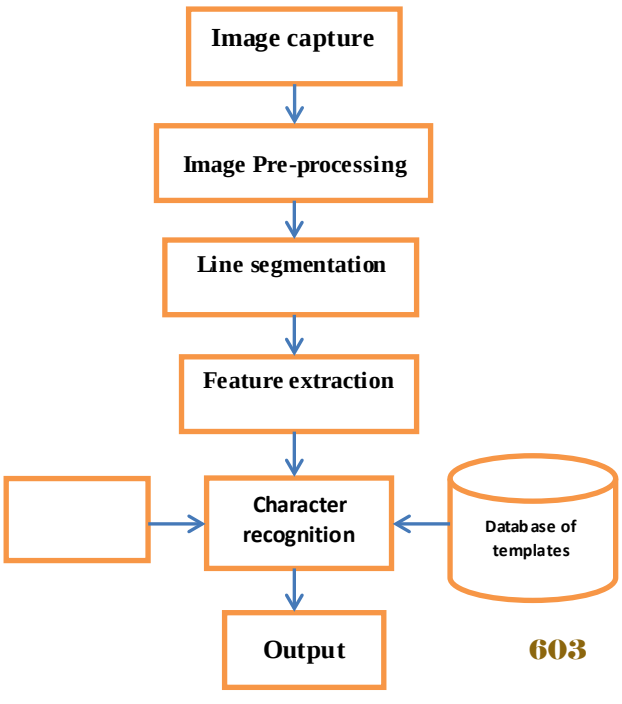


Fig-1: The proposed structure for Arabic handwritten character recognition.

The method contains sequences of steps, each stage passing its results to the next stage as shown in fig.1; feedback loop is not needed in the process. Once the input data is captured and stored, the text image goes through image preprocessing, line segmentation, feature extraction, character recognition stages. In this section, details of each stage will be given.

3.1. Image Capture

Image capture is the first stage, and provides an input to the system. The major job of image capture module is to get text image. It is called 'image' since scanner intrinsically scans pixel of the text and not characters.

The image captures module acquires handwritten Arabic characters by a digital camera. The digital camera, which is utilized, is culminated in its capability, and accuracy rate. It works many services depend on the information that we need to filter. The capacity of the digital camera is to perceive the input data. The system offers a work space to store image capture in file such as jpg, bmp etc., and the procedure passes to the next stage.

3.2. Image Preprocessing

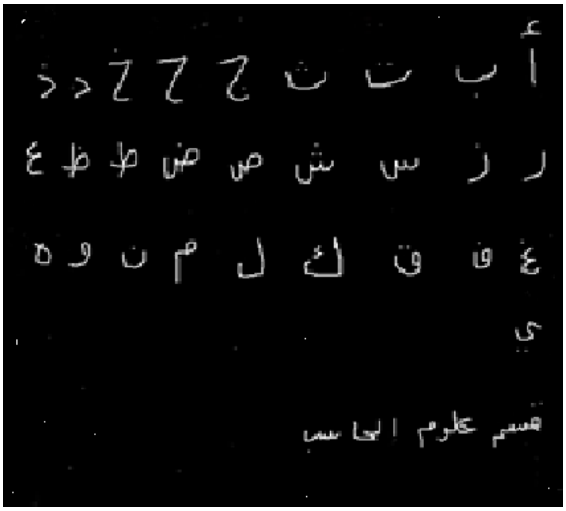
The image Preprocessing is an alluring stride to: enhance the execution, diminish varieties, and create a more predictable arrangement of data preceding computational running. This step has four sub-procedures, which are: 1) gray scaling, 2) image binarization, 3) image noise removal, 4) image cropping and resizing and 5) morphological operations. Details of every sub-process will be given in this section.

3.2.1. Gray Scaling

In this phase, input image is changed into gray scale image. Gray scale image is an image in which each pixel holds intensity information, black at lowest intensity and white at highest.

3.2.2. Image Binarization

Image binarization is the process of altering a gray scale image (0 to 255 pixel values) to binary image (0 and 1 pixel values). In this stage, gray scale image is transformed into a binary image. These changes over the gray scale image into black and white image, where the pixel values of the image are either 0 or 1 (binary image). The image binarization process converts an inserted character image into a binary image (0s and 1s only). Digital image is defined as a



two-dimensional object with a finite set of intensity values whose elements are referred as image elements (pixels). Images whose possible intensity values are only black (foreground) represented as 1 and white (background) represented as 0. Figure 2 demonstrates a sample image after black & white processing. This makes pixels modeling the region are categorized into two kinds: foreground indicating the text, and background indicating blank regions. From a gray scale image, thresholding method is utilized to make binary images. Thresholding generates binary images from gray level by turning all pixels beneath some threshold to zero and all pixels about that threshold to one. In the event that $g(x, y)$ is a thresholded form of $f(x, y)$ at some global threshold T,

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

Pixels which have value 1 are related to the object, and pixels which have value 0 are corresponded to background (see Fig. 2). T is a constant; this approach is called global thresholding. Global thresholding method is operated to change a gray level image to a binary image.

Figure 2 A sample handwritten Arabic characters image after black & white processing. Obviously, there is noise in the sample.

3.2.3. Image Noise Removal

Arabic handwritten Character images are inclined to an assortment of sorts of noise. Noise can be presented into an image. It produced by the digitizer or by a shaking hand. The Arabic characters have dots situated at the top or bottom of their primary body making them helpless against noise. To eliminate noise from the image, duplicated data points are removed by compelling a least gap between successive points. The initial phase in image noising removal is to decide if the pixel that is being handled is noisy or not. In this paper, noise is

expelled by utilizing the function Medfilt2 in Matlab r2017a. B=medfilt2(A, [mn]) operates average filtering of the matrix A in two dimensions, where each output pixel includes the average value in the m-by-n neighborhood nearby the matching pixel in the input image.

3.2.4. Image Cropping and Resizing

When the image is free from noise, the added portion present in the image other than the portion taken by the character should be dispensed so that only the character is processed. This procedure is called cropping. To crop an image, the top-leftmost black pixel, the top-rightmost black pixel, bottom-leftmost black pixel, and bottom-rightmost black pixel of the character are found and kept. These values are parameters to the cropping procedure to remove just the character from the image. Once cropping the character image is done, the image is resized to a standard size of 16*16. Resizing procedure determines number of rows and columns, which are set for any character image. So, each entered image is resized to 16 x 16 pixels image. This is a pattern size for all the images to be studied.

3.2.5. Morphological Operations

To continu effortlessly for further handling, the character in the image should be improved. This can be accomplished by using dilation, an operation in mathematical morphology.

The dilation operation as a rule uses an organizing component for examining the shapes contained in the input image. The effect of the dilation operation on a binary image is to expand the boundaries of regions of white pixels (foreground). Therefore, areas of foreground pixels increased in size whereas holes in those areas become smaller. Two functions are proposed, *morph* function, and *dilation* function. The *morph* function is utilized before execution *dilation* function to generate a morphological structuring component of the sort determined by shape and radius. The result of *morph* function is delivered to the *dilation* function together with the image to be dilated. Therefore, the output of the *dilation* function improves character image for the next stage. Figure 3 shows a sample image after noise removal and morphological filtering processes.

Figure 3 A sample handwritten Arabic characters image after noise removal and morphological filtering processes.

3.3. Line Segmentation

The input document might include many lines of text that must be sorted. Between any two lines of text, there are many horizontal spaces with either white pixel or very little black pixels. Therefore, these horizontal spaces are scanned for break points through them and they will be stored. To segment a line, the input document is scanned for the first black pixel (f) and the last black pixel (b) until get white horizontal spaces. The distance between (f) and (b) gives the height (h) of the text line. The area (row) between (f) and (b) with the height (h) contains all pixels of the characters in the line image. These pixels are stored. Thus, lines of handwritten image are ready for classifying.

3.4. Feature Extraction

After line character segmentation process is finish, text line rows are retrieved from right to left to get features of every Arabic character. Feature Extraction depicts the geometrical and topological attributes of a text that might be word, character, stroke, or digit by depicting its global and local properties. Features rely on the character of text that needs to be grouped. For Arabic text, these features contain loops, dots, cross points, strokes, and branch points in various ways. A method is implemented for feature extraction based on structural features to get the exactness shape of a connected pixels and component of each character as well as horizontal and vertical projection drawings in 2-D binary image. Connected component is an important feature because most of Arabic characters contain one or more connected components like (أ, ب, ت, ث, ج, خ, ذ, ز, ش, ض, غ, ض, ظ, ق, ش, ك, ف, ي).

Searching starts from right to left for finding connectivity of Arabic character and label connected components. The first black pixel found is the rightmost (right) pixel of character, and in the event that all pixels are observed to be white then this is right of character. The separation of the word into characters is done by operating blob analysis. Each character is named as a blob. Blob Analysis is a key system of machine vision based on testing of regular image areas and splits a word into characters, each character is termed blob. A blob contains a collection of connected pixels. The idea is to trim out each labeled connected components of pixels by finding its minimum and maximum values of its row and column and extracting the character out. Locating connected components is done in the column-wise mode (i.e. from top to bottom scan order). This is done by refining the extracted characters to fit them into a window without white areas on all the four sides and generating the template for each extracted character. The templates are standardized to 16x16 pixels and kept in a database. Normalization is performed utilizing window to view port conversion.

This mapping is utilized to map every pixel of the initial image to the matching pixel in the standardized image. In the next stage, the extracted character will be used as input to the character classifier, and it will be matched with all the characters in the database to classify similarity.

3.5. Character Recognition

3.5.1. Template Matching

Template matching is character recognition method that locates the area of a sub image called a template inside an image. After various relating templates are found, their centers are utilized as conforming points to settle the registration parameters. Template matching includes deciding similitude between a given template and windows of a similar size in an image and finding the window that delivers the most astounding likeness measure. It operates by contrasting determined image features of the image and the template for every conceivable dislodging of the template. This procedure includes the utilization of a database of templates.

3.5.2. Template Matching Algorithm

An algorithm is proposed for template matching based on locating areas of an input image that match to a template image. All image pixels are utilized as features. To recognize the matching area, the template image is compared against the input image. Comparison results between the input character and the template are measured for similarity, using 2-D normalized cross correlation coefficient. The algorithm stages are as follows :

Stage 1: Load an input text image and a patch image (*template*).

Stage 2: Move the template image above the input text image.

Stage 3: Image is tested according to the pixel size (left to right, up to down).

Stage 4: Compare similarities between a certain group of templates and an input image.

Stage 5: Calculate the 2-D normalized cross correlation coefficients from every comparison between the input image and the template, to find similarities.

Stage 6: Determine the template that gives the maximum similarity (matching region).

Stage 7: Convert the recognized character images to text format.

This procedure involves the use of a database of templates that are standardized to 16x16 pixels. There exists a template for all likely input characters. For recognition to happen, the recent input character is compared with each templates using 2-D normalized correlation coefficients method to find maximum similarity between an input image and the standard database images. The common similarity measure applied in practice is a *2-D normalized cross-correlation* function, performed by the following equation J. P. Lewis [19]:

$$f(u,v)=\frac{\sum_{x,y}[I(x,y)-\bar{I}_{u,v}][T(x-u,y-v)-\bar{T}]}{\sqrt{\sum_{x,y}[I(x,y)-\bar{I}_{u,v}]^2\sum_{x,y}[T(x-u,y-v)-\bar{T}]^2}} \quad (1)$$

Where f is the 2-D normalized correlation coefficient, I is the input image, T is the template image, $\bar{T}$ is the mean of the template, and $\bar{I}_{u,v}$ is the mean of $I(x,y)$ in the area below the template. The template T is moved *u* steps in the *x* direction and *v* steps in the *y* direction of the input image I , and then the comparison is computed over the template region for each location (*u*,*v*). The value of 2-D normalized cross-correlation coefficient ranges from -1 to +1, totally not matched and totally matched respectively.

After all characters and words recognized, the output image are converted into Arabic character's language format. A .doc file generated and the output image stored in the file. Every character grouped into word and the output is displayed as shown in Figure 4.

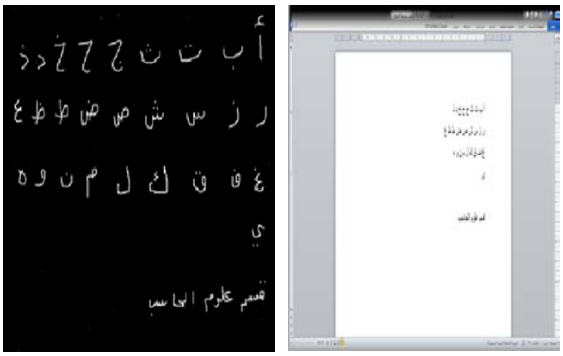


Figure 4 A text image sample and its output.

4. Implementation and Experimental Results

No standardized test sets exist for character recognition, and as the performance of an OCR system is highly dependent on the quality of the input, this makes it difficult to evaluate and compare different systems [1]. Still, recognition rates are often given, and usually presented as the percentage of characters correctly classified [1]. To examine the proposed method and determine its capability to recognize input Arabic handwritten character, Arabic handwritten characters taken randomly from 20 persons.

Each person has written 28 Arabic characters from أ through ي . Generally, the tried characters are about 560 separate characters extending from أ to ي. Each character has been revised 20 times originating from different persons. In an evaluation of OCR system, three different performance rates should be investigated [1]: error rate, rejection rate, and recognition rate. Error rate (ER) consequences from the forged template that is accepted by the system erroneously amid testing. Rejection rate (JR) results from the veritable template that the system recognizes as the inquiry template wrongly. Finally,

recognition rate (RR) is the overall successful accuracy.

In the system of this paper, the (ER) rate does not exist because there are no forged templates. Hence, (ER) rate is mostly estimated to be zero. However, (JR) is mainly utilized for the testing measure to evaluate the recognition percentage, because some of an Arabic handwritten character can be veritable templates, therefore, the system was unable to recognize it. (JR) accuracy is equal to the number of characters that cannot be recognized by the system to the total number of characters in an image. The overall successful accuracy (RR) is equivalent to the number of characters recognized by the system to the total number of characters in an image. The experimental results indicated that the recognition error measure in terms of (JR) is 3 %, and the overall successful accuracy (RR) is 97 %.

5. Conclusion

A simple and effective template matching technique for Arabic handwritten character recognition was presented in this paper. The system inserts Arabic handwritten characters (text) image, preprocesses the image, extracts appropriate image features, classify the characters, and recognizes the image. After preprocessing stage finished, the characters were extracted from an input image and normalized. For recognition process, the extracted character was matched with each template in the database to find the closest image of the input character. The matching metric was computed using 2-D correlation coefficients approach to identify similar patterns between the test image and the database images. The proposed system can be used to generate databases of existing Arabic handwritten text without utilizing the keyboard. Hence, It increases the speed of input process, decreases possible human errors, and allows compact storage and file operations. Experimental results prove that the proposed system is efficient for recognition of Arabic handwritten characters. The proposed system shows the recognition effectiveness all alone test dataset with a general precision of 97% for Arabic handwritten characters. In the future work, I will attempt to enhance the template matching methodology by joining it with the neural network system to mollify higher identifying rate for handwritten Arabic character recognition.

References

[1] Jagruti Chandarana, Mayank Kapadia, Optical Character Recognition, International Journal of Emerging Technology and Advanced Engineering, Vol. 4, No. 5, 2014, pp. 219-223.

[2] Najib Ali Mohamed Isheawy And Habibul Hasan, Optical Character Recognition (OCR) System, IOSR Journal of Computer Engineering, Vol. 17, No. 2, 2015, pp. 22-26.

[3] Schantz, Herbert F. (1982). The history of OCR, optical character recognition, Manchester Center, Vt. : Recognition Technologies, Users Association. ISBN 9780943072012

[4] Faisal Mohammad, Jyoti Anarase, Milan Shingote, Pratik Ghanwat, Optical Character Recognition Implementation Using Pattern Matching, International Journal of Computer Science and Information Technologies, Vol. 5, No. 2, 2014, pp. 2088-2090.

[5] Parvez, M.T., Sabri, A.M., 'Arabic handwriting recognition using structural and syntactic pattern attributes', Pattern Recogn., Vol. 46, No. 1, 2013, pp. 141-154.

[6] Usman Saeed, Automatic Recognition of Handwritten Arabic Text: A Survey, Life Science Journal, Vol. 11, No. 3s, 2014, pp. 232-235.

[7] A, Mars. & G, Antoniadis ,Handwriting recognition system for Arabic language learning, WCITCA'2014 World Congress on Information Technology and Computer Application, HAMMAMET 2015 – International Journal N&N Global technology.

[8] Majid Ziaratban, Karim Faez, Farhad Faradji, Language based feature exraction using template matching in Farsi/Arabic handwritten numeral recognition, Ninth International Conference on Document Analysis and Recognition, vol. 1, 2007, pp. 297–301.

[9] Sunny Kumar, Pratibha Sharma, Offline Handwritten & Typewritten Character Recognition using Template Matching, International Journal of Computer Science & Engineering Technology, Vol. 4 No. 06, 2013, pp. 818-825.

[10] Nikhil Rajiv Pai, Vijaykumar S. Kolkure, Design and implementation of optical character recognition using template matching for multi fonts /size, International Journal of Research in Engineering and Technology, Vol. 4, No. 2, 2015, pp. 398-400.

[11] Mo Wenying, Ding Zuchun, A Digital Character Recognition Algorithm Based on the Template Weighted Match Degree, International Journal of Smart Home, Vol. 7, No. 3, 2013, pp. 53-60.

[12] Jatin M Patil, Ashok P. Mane, Multi Font And Size Optical Character Recognition Using Template Matching, International Journal of Emerging Technology and Advanced Engineering, Vol. 3, No. 1, 2013, pp. 504-506.

[13] Soumendu Das, Sreeparna Banerjee, An Algorithm for Japanese Character Recognition, International Journal of Image, Graphics and Signal Processing (IJIGSP), Vol. 7, No. 1, 2014, pp. 9-15.

[14] Md. Mahbubar Rahman, M. A. H. Akhand, Shahidul Islam, Pintu Chandra Shill, M. M. Hafizur

Rahman, Bangla Handwritten Character Recognition using Convolutional Neural Network, International Journal of Image, Graphics and Signal Processing (IJIGSP), Vol. 7, No. 8, 2015, pp. 42-49.

[15] N. Shobha Rani, Vasudev T, and Pradeep C.H, A Performance Efficient Technique for Recognition of Telugu Script Using Template Matching, I.J. Image, Graphics and Signal Processing, Vol. 8, 2016, pp. 15-23.

[16] Seema Barate, Chaitrali Kamthe, Shweta Phadtare, Rupali Jagtap, M.R.M Veeramanickam, Text Character Extraction Implementation from Captured Handwritten Image to Text Conversion using Template Matching Technique, MATEC Web of Conferences, Vol. 57, No. 01010, 2016, pp. 1-6.

[17] Kishori Kokate, Surekha Thube, Tejashree Kautkar, Data Entries and Location Indication of Product Using Arduino, IJCSN International Journal of Computer Science and Network, Vol. 6, No. 2, April 2017, pp. 72-75.

[18] Kajal Gade,Madhuri Mogal,Heena Sindhani,Prajakta Sonawane,Vaishali Khandve, DSP based Optical Character Recognition for Devnagari charaters, IJARIIE, Vol. 3, No. 2, 2017, pp. 2774-2778.

[19] J.P. Lewis, Fast normalized cross-correlation, in: Proceedings of Vision Interface, 1995, pp. 120-123.