Research Article

Ancient Geez script recognition using deep learning

Fitehalew Ashagrie Demilew[1] · Boran Sekeroglu[2]

Abstract
Handwritten text recognition is one of the most valuable recognition systems because of the unique characteristics of each person's handwriting. Thus, recognition systems need to be more adaptable to recognize same or different characters with different characteristics. On the other hand, one of the most challenging tasks in handwritten text recognition problems is recognizing ancient documents which include several noise within them. While digitizing these documents, these noise appear in different types which effects any recognition system. So, digitizing ancient documents, applying proper pre-processing techniques and performing effective classifier are the main steps of efficient recognition system. In this paper, a complete Ethiopian ancient Geez character recognition system using deep convolutional neural network is proposed in order to recognize twenty-six base characters of this alphabet. The proposed system obtained an accuracy of 99.39% with a model loss of 0.044 which demonstrates its efficiency.

Keywords Handwritten character recognition · Ancient document recognition · Convolutional neural networks

1 Introduction

Ethiopia is the only African country with its own indigenous alphabets and writing systems which is the Geez or Amharic alphabet [1]. Most of the African countries use English and Arabic scripts or alphabets.

Geez is a type of Semitic language and mostly used in Ethiopian and Eritrean Orthodox Tewahedo churches (EOTC). Geez belongs in the South Arabic dialects and Amharic which is one of the most spoken languages of Ethiopia [2]. There are more than 80 languages and up to 200 dialects spoken in Ethiopian; some of those languages use Geez as their writing script. Among the languages, Geez, Amharic, and Tigrinya are the most spoken languages, and they are written and read from left-to-right, unlike the other Semitic languages [3]. There are lots of ancient manuscripts which are written in Geez currently in Ethiopian, especially in the EOTCs. However, it is not possible to find a digital format of those manuscripts due to a lack of optical character recognition (OCR) systems that can convert them.

OCR is the process of extracting characters from an image and converting each extracted character into an American Standard Code for Information Interchange (ASCII), Unicode, or computer editable format. Handwritten character recognition (HCR) involves converting a large number of handwritten documents into a machine-editable document containing the extracted characters in the original order. Therefore, technically the main steps of handwritten text recognition are image acquisition, pre-processing, segmentation, feature extraction, classification, and/or possibly post-processing [4].

Generally, there are two types of handwriting text recognition systems which are offline and online text recognition. Online text recognition is applied to data that are captured at present or real-time. For online text recognition, information like pen-tip location points, pressure, and current information while writing is available which are not available in the case of offline recognition. Thus, online

✉ Boran Sekeroglu, boran.sekeroglu@neu.edu.tr; Fitehalew Ashagrie Demilew, 20176813@std.neu.edu.tr | [1]Department of Software Engineering, Near East University, Cyprus, Mersin 10, Turkey. [2]Department of Information Systems Engineering, Near East University, Cyprus, Mersin 10, Turkey.

SN Applied Sciences (2019) 1:1315 | https://doi.org/10.1007/s42452-019-1340-4

Received: 14 August 2019 / Accepted: 24 September 2019 / Published online: 1 October 2019

text recognition is easy when it is compared to offline recognition [5]. In offline text recognition, a scanned image or image captured using a digital camera is used as an input to the software to be recognized [6]. Once the images are captured, some pre-processing activities are applied to them in order to recognize the characters accurately. Hence, offline recognition requires pre-processing activities on the images, and it is considered as a difficult operation than online recognition.

There are many different kinds of classification methods which have been proposed in different researches [7–9]. Every approach has its own advantages and disadvantages, so it is difficult to select a general and efficient classification approach [10]. Many researchers proposed different systems or approaches in order to improve the efficiency of the recognition process. HCR involves many relevant phases or stages like data acquisition, pre-processing, classification, and post-processing. Each phase has its own objectives, and its efficiency defines the accuracy of the next phases and finally the overall recognition process.

Ancient script or document recognition is far harder than the printed character recognition and handwritten character recognition processes because of aging, staining, ink quality, etc. [11]. In [12], the researchers conducted an ancient Devanagari document recognition using different kinds of character classification approaches. The approaches used by the researchers are ANN, fuzzy model, and support vector machine (SVM). They have achieved an accuracy of 89.68% using a neural network classifier and 95% accuracy using a fuzzy model classifier for the numerals and 94% using SVM and 89.58% using multilayer perceptron (MLP) for the alphabets.

There are few studies conducted about Ethiopic document recognition [13] and Geez manuscript recognition [14], but these researches did not consider ancient Geez document recognition.

In this paper, we have proposed an offline ancient Geez document recognition system using deep convolutional neural network in which the convolutional neural network integrates an automatic feature extraction and classifications layers. The proposed system includes the pre-processing stage of digitized ancient images, the segmentation stage to extract each character, and the feature extraction within the convolutional neural network architecture.

2 Image preparation

In general, many character recognition systems either handwritten or printed recognition follow common steps with a little variances with some exceptional researches [15, 16]. These variances may include a difference in the processes, techniques, and implementation strategies. In the proposed system, basic steps are followed during the development phase with some relevant changes and modifications. General diagram that shows the most common steps of character recognition systems and the proposed system can be seen in Fig. 1.

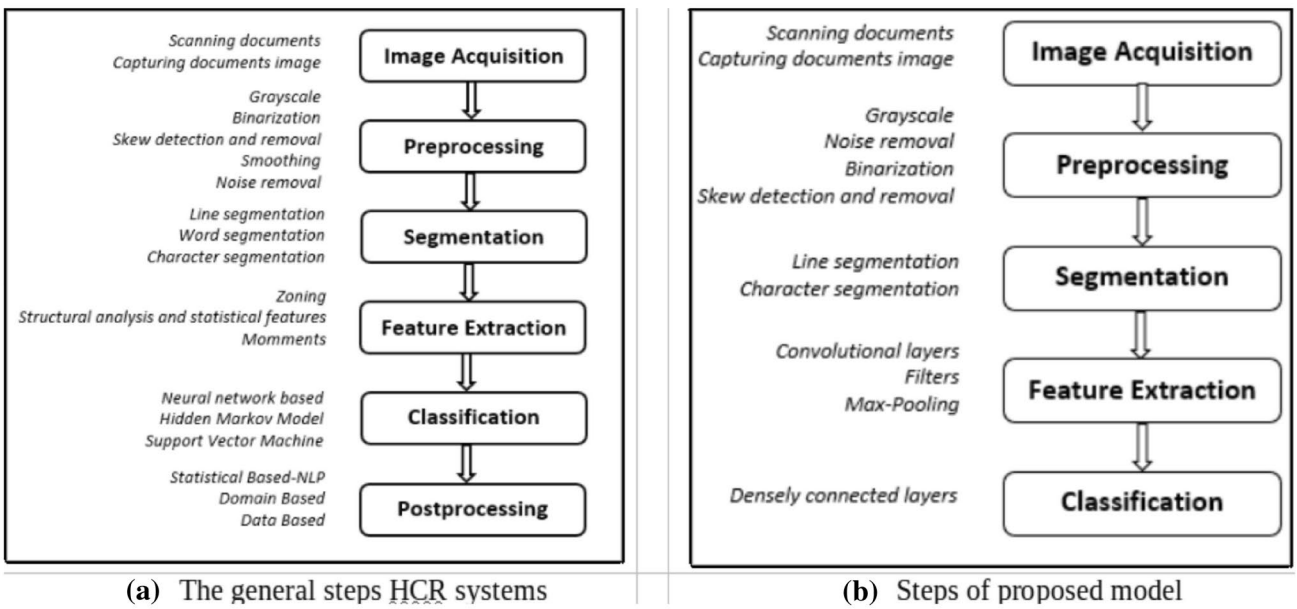

Fig. 1 Steps of handwritten character recognition systems

2.1 Image acquisition and pre-processing

Image acquisition phase is the first step for any image recognition system, also for character or text recognition systems. Digital scanner can be an effective tool for digitizing documents with high resolution. However, it is impossible to use digital scanners in ancient documents, while touching is forbidden to these documents because it can be harmful to them. Thus, digital camera is used to capture and digitize ancient documents.

During this process, several types of problems occur on the images and thus they should pass through the pre-processing stages. The purpose of these stages is to remove irrelevant pixels to form a better representation of the images and make them ready for the subsequent stages. This stage involves different sub-processes such as grayscale conversion, noise reduction, binarization, smoothing, and skew detection and correction.

2.1.1 Grayscale conversion and denoising

Captured RGB images are converted to grayscale in order to decrease the intensity dimension of image. This provides more efficient input data for post-steps and decreases computational time which is one of the common steps in all character or text recognition steps.

After grayscale conversion, it is an essential process to apply denoising step while dealing with handwritten document recognition, especially conducting ancient document recognition, since the documents are highly noise and degraded. After several experiments, most accurate results are obtained by non-local means denoising [15]. It works by finding the average of similar pixels and updating the pixel value with the calculated average pixel value. The whole image is scanned for finding pixels having similar intensity with the pixel to be denoised. Denoising is basically done by computing this average value of the pixels having similar values. The similarity is obtained by Eq. 1.

$$NLu(p) = \frac{1}{C(p)} \int f(d(B(p), B(q)u(q)\mathrm{d}q)) \tag{1}$$

where $d(B(p), B(q))$ is an Euclidean distance between p and q in which they are patches of the image. Also, $C(p)$ and f are the normalization and decreasing functions, respectively, and $B(p)$ and $B(q)$ are representing neighborhoods centered at p and q, a pixel size of $(2r + 1) \times (2r + 1)$.

Figure 2 presents original image, grayscale conversion and denoising operation, respectively.

2.1.2 Binarization and skew detection

After denoising process, it is required to binarize document image in order to prepare it for segmentation or feature extraction process. This is one of the vital part of document recognition that effects recognition efficiency of the system.

Several binarization methods have been proposed as global [17, 18] and local [19–21] to enhance or to segment document images. It is a crucial part to determine which method is superior, and different researches have been conducted to determine superior method for document images [22]. But it is mentioned that local methods have more disadvantages than global ones, especially because of the determination of their kernel size. Small kernel size adds additional noise, and large kernel size acts as global methods. Thus, global methods are preferred in document image binarization [23]. Commonly, Otsu method [24] was suggested as the most stable and accurate method and determined to be used in this step of proposed system [22].

The Otsu method is intended to find a threshold value that can separate the foreground of the image from

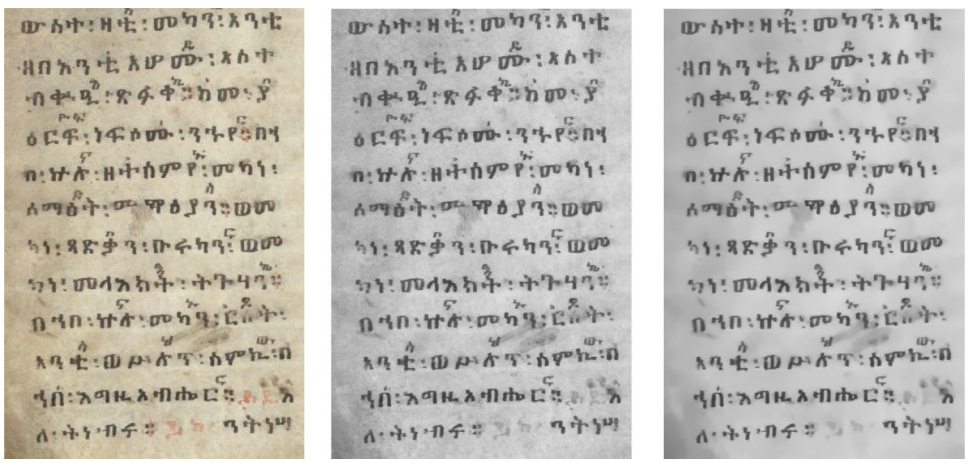

Fig. 2 Original image, grayscale conversion, and denoising process

background so that it can minimize the overlap that occurs between the white and black pixels. It used discriminant analysis to divide the foreground and background.

Then, image negatives are applied in order to get the characters' pixels as white and the background as black.

Once the image is denoised and binarized, the skew detection and correction method based on Hough transform [25] is applied to the binary image. Skew detection is required because of the irregular angle of the digital cameras during the image acquisition process. Initially, the algorithms find all the pixel coordinates, which are part of the foreground. The foreground pixels are all the points having a pixel color of white. Then, the rotation of the rectangle is calculated based on the collected pixel coordinates. The rotation angle is between − 90 and 0. By using the rotation, matrix can be found with respect to the center coordinates of the image through Eq. 2.

$$\begin{bmatrix} \alpha & \beta & (1-\alpha) * \text{center}.x - \beta.\text{center}.y \\ -\beta & \alpha & \beta.\text{center}.x + (1-\alpha).\text{center}.y \end{bmatrix} \quad (2)$$

where $\alpha = \cos r, \beta = \sin r$.

2.1.3 Segmentation

Segmentation is a critical step in the handwritten recognition process, and it highly affects the recognition process. Usually, the segmentation process involves three steps which are line segmentation, word segmentation, and character segmentation. For each segmentation process, contour analysis plays a greater role in the proposed system. Basically, for line segmentation and character segmentation, an special technique of adding additional foreground colors horizontally and vertically is used. In general, filling additional pixel is also called dilation and this technique is used for line segmentation. However, the word segmentation is not conducted in the segmentation process because of the nature of Geez documents. The methods used in the proposed system are described as follows.

Text line segmentation is an essential step in which every line is cropped out of the image and stored in a sorted manner. Thus, using those lines words and characters can be extracted from each and every line with the original order. Contour locating algorithm [26] is used for finding contours in both line and character segmentation processes. Figure 3 presents binarization, skew detection, and line detection processes on denoised image.

Contour locating algorithm uses border tracing technique to find all contours or objects on the image. The algorithm finds the borders of each contour and returns their location coordinates. The coordinates returned are the points of the top-left, top-right, bottom-left, and bottom-right corners of each contour. Then the coordinates are used to process the contours or objects. The steps that the algorithm follows for finding contours are listed below.

- Scan the binary image while updating *P*, where *P* is a counted sequential number of the last found outer most border.
- Represent each hole by shrinking the pixels into a single pixel based on a given threshold value *t*, where *t* is a minimum perimeter of a character or line in line segmentation.
- Represent the outer border by shrinking its pixels into a single pixel based on a given perimeter or threshold value *t*.
- Place the shrink outer border pixel to the right of the shrank hole pixel.
- Extract the surrounding relations connected components, i.e., holes and outer borders.

In Geez, literary words are not separated with spaces, rather with a punctuation mark. The word separator has

Fig. 3 Binarization, skew detection, and line detection processes

a colon-like structure, so each word is separated with a colon from another word. Since the system can recognize some of the punctuation marks, the word separator will be recognized in the classification stage. One of the aims of recognizing some of the punctuation marks was word segmentation. Thus, there is no need for word segmentation.

Once the lines are segmented successfully, character segmentation is carried out. In the proposed system, again contour analysis [26] is used. Figure 4 presents line and character detection on original image.

3 Feature extraction and classification

After the segmentation process, the system has all of the isolated characters and ready to extract the features that will uniquely represent the characters.

Convolutional neural networks are composed of two parts. The first part contains a number of convolutional layers which are a special kind of neural network that is responsible for extracting features from the input image, and the second part includes dense neural layers which are required to classify the features extracted from the convolution layers.

On the proposed system, three convolutional layers are used for feature extraction and two dense layers for classification purpose. The designed convolutional neural network architecture has a number of convolutional layers, filter kernels, activation function, fully connected layers, and pooling specifications as specified in Table 1.

The output of the second convolutional network is the features, and they are flattened. Thus, they can be passed to the fully connected or dense layers. In Table 2, specifications of the dense layers that are responsible for using and classifying the features extracted from the convolutional layers can be seen.

General block diagram of deep CNN can be seen in Fig. 5.

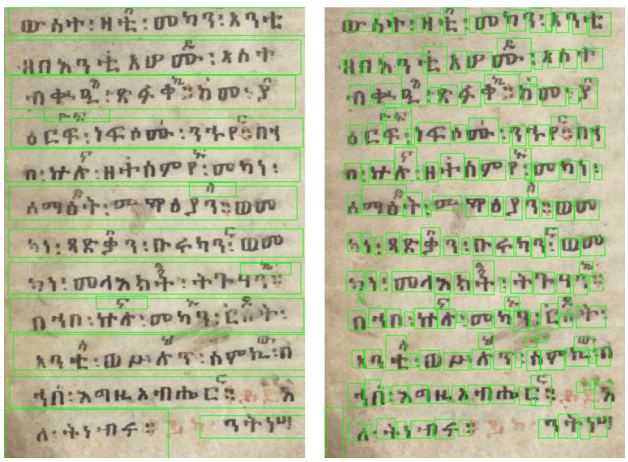

Fig. 4 Line and character detection on original image

4 Experimental results

In this section, dataset and performed experiments will be explained in details.

4.1 Dataset

For training and testing, a dataset has been prepared that contains the base characters and some of the punctuation marks of the Geez alphabet. The dataset is prepared from a total of 208 pages collected from EOTCs, libraries, and private books. The dataset contains 22,913 binary image characters which are resized into 28*28 pixels. However, the frequency of the extracted characters is varied. The frequency varies from character to character and the minimum number of images per class is around 400 and the maximum is 1700. Figure 6 demonstrates the most frequently used characters.

Finally, the prepared dataset is separated into three parts as training, testing, and validation sets. 70% of dataset which is 16,038 characters of 22,913 is used for training, 20% for testing (4583 characters), and 10% (2292 characters) for validation.

Table 2 Deep layer specifications of the system

Layers	Dropouts	Input shape	Activation function
Dense1 layer	0.2	256	ReLU
Dense2 layer	0.2	256	ReLU
Output layer	Non	28 (number of classes)	SoftMax

Table 1 Convolutional layer specifications of the system

Layers	Input size	Filters	F-size	Pooling	P-size	Dropout	Activation
Conv1	28*28	32	5*5	Max-pooling	2*2	0.25	ReLU
Conv2		64	3*3	Non	Non	0.2	ReLU
Conv3		128	2*2	Non	Non	Non	ReLU

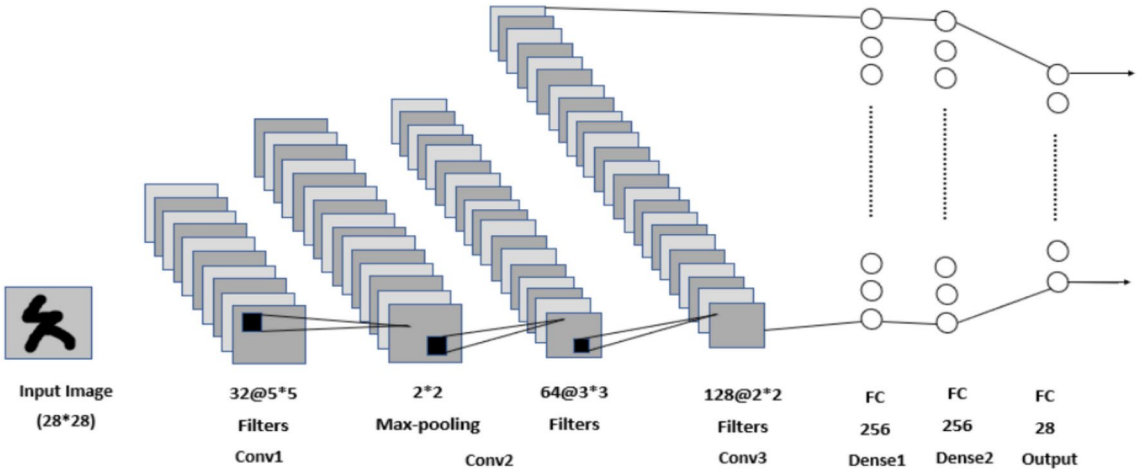

Fig. 5 Convolutional neural network architecture of the proposed system

Characters	Frequency
ዐ	1396
ፀ	1108
አ	1613
ቨ	1005
ሀ	1091
ዘ	1356
፡	1692

Fig. 6 Most frequently used characters in Ethiopian ancient documents

4.2 Results

For line and character segmentation, the applied algorithm accepts the binary images that have passed through all the pre-processing stages. The algorithm has been tested by a number of documents images, and its accuracy was as expected. However, in some cases when there is a connection among the lines the algorithm was unable to segment the lines accurately. Basically, the connection that exists among the lines was supposed to be removed on the noise removal and morphological transformation stages.

After training the CNN with the prepared dataset, it is observed that when the epoch value increases, the accuracy of the system is improved. However, when the epoch value is raised to 20, improvement in the convergence of model starts to decrease. The highest results obtained in 15 epochs with 0.993890 and 0.04442 test accuracy and test loss, respectively, and with 0.9895 and 0.0773 validation accuracy and validation loss, respectively, by 28 mis-classified characters out of the total 4583 testing images. Table 3 presents obtained results of the system with different epochs. Bold values demonstrate the optimum results obtained.

Table 3 Obtained results of the proposed system

Eopch	Test accuracy	Test loss	Validation accuracy	Validation loss
10	0.992363	0.04781	**0.9895**	0.0756
15	**0.993890**	**0.04442**	**0.9895**	0.0773
20	0.991708	0.05239	0.9887	**0.0508**

4.3 Comparisons

It is excessively difficult to find related research on ancient Geez document recognition. One of the rarely conducted research about this subject is by Siranesh and Menore [14]. Therefore, obtained results are compared with this results obtained in the their research. Table 4 shows the comparison results of the proposed system and other research.

5 Conclusions

Ancient document recognition has a variety of challenging problems such as noisy images, degraded quality, types and characteristics of ink and digitization processes. Thus, effective pre-processing phase and adaptive classification is required to obtain superior results than other researches. This paper focused on the development of ancient Geez document recognition system using a deep convolutional neural network. The proposed system involves pre-processing, segmentation, feature extraction, and

Table 4 Comparison of the results with related research

	Test accuracy (%)	Test loss	Epoch	Model
Highest of proposed system	**99.38**	**0.04442**	15	Deep CNN
Highest of Siranesh and Menore	93.75	0.062	150	Deep NN
Lowest of proposed system	99.17	0.052	20	Deep CNN
Lowest of Siranesh and Menore	92.0	0.218	50	Deep NN

classification stages. A dataset was prepared and used for training and testing of the proposed system.

Most of the documents were difficult even to recognize through naked eyes; however, the proposed system obtained an optimal accuracy with applied pre-processing steps by 99.389% and loss of 0.044. This proves that the proposed system can be an effective way for document recognition and particularly for ancient documents.

More comprehensive dataset may increase this accuracy, and future work will include this with implementation of other deep learning and classification models.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Sahle E (2015) Top ten fascinating facts about Ethiopia Media has not told you. Retrieved 2 Jan 2019 from https://africabusiness.com/2015/10/29/ethiopia-4/
2. Britannica (2015) The editors of encyclopedia, Geez language. Retrieved 21 March 2019 from http://www.britannica.com/topic/Geez-language-accordion-article-history
3. Bender M (1971) The languages of Ethiopia: a new lexicostatistic classification and some problems of diffusion. Anthropol Linguist 13(5):165–288
4. Kim G, Govindaraju V, Srihari SN (1999) An architecture for handwritten text recognition systems. Int J Doc Anal Recognit 2(1):37–44
5. Ahmad I, Fink G (2016) Class-based contextual modeling for handwritten Arabic text recognition. In: 2016 15th international conference on frontiers in handwriting recognition (ICFHR)
6. Shafii M (2014) Optical character recognition of printed Persian/Arabic documents. Retrieved 16 Dec 2018 from https://scholar.uwindsor.ca/etd/5179
7. Kavallieratou E, Sgarbas K, Fakotakis N, Kokkinakis G (2003) Handwritten word recognition based on structural characteristics and lexical support. In: 7th international conference on document analysis and recognition
8. Nasien D, Haron H, Yuhaniz SS (2010) Support vector machine (SVM) for English handwritten character recognition. In: 2010 2nd international conference on computer engineering and applications
9. Pradeep J (2012) Neural network based recognition system integrating feature extraction and classification for English handwritten. Int J Eng 25(2(B)):99–106
10. Cheriet M (2007) Character recognition systems: a guide for students and practitioners. Wiley Interscience, Hoboken, pp 32–34
11. Laskov L (2006) Classification and recognition of neume note notation in historical documents. In: International conference on computer systems and technologies
12. Malanker A, Patel P (2014) Handwritten Devanagari script recognition: a survey. IOSR J Electr Electron Eng 9(2):80–87
13. Bigun J (2008) Writer-independent offline recognition of handwritten Ethiopic characters. ICFHR 2008
14. Siranesh G, Menore T (2016) Ancient Ethiopic manuscript recognition using deep learning artificial neural network (Unpublished master's thesis). Addis Ababa University
15. Yousefi M, Soheili M, Breuel T, Kabir E, Stricker D (2015) Binarization-free OCR for historical documents using LSTM networks. In: 2015 13th international conference on document analysis and recognition (ICDAR)
16. Pal U, Wakabayashi T, Kimura F (2007) Handwritten Bangla compound character recognition using gradient feature. In: 10th international conference on information technology (ICIT 2007)
17. Kittler J, Illingworth J (1986) Minimum error thresholding. Pattern Recognit 19(4):4147
18. Esquef IA, Mello ARG, de Albuquerque MP, de Albuquerque MP (2004) Image thresholding using Tsallis entropy. Pattern Recognit Lett 25:10591065
19. Sauvola J, Pietikainen M (2004) Adaptive document image binarization. Pattern Recognit 33:225236
20. Kim IK, Jung DW, Park RH (2002) Document image binarization based on topographic analysis using a water flow model. Pattern Recognit 35:265277
21. Chen Y, Leedham G (2005) Decompose algorithm for thresholding degraded historical document images. IEE Proc Vis Image Signal Process 152(6):702714
22. Sekeroglu B, Khashman A (2017) Performance evaluation of binarization methods for document images. In: Proceedings of the international conference on advances in image processing, pp 96–102
23. Khashman A, Sekeroglu B (2007) A novel thresholding method for text separation and document enhancement. In: 11th Panhellenic conference in informatics, pp 323–330
24. Otsu N (1979) A threshold selection method from gray-level histograms. IEEE Trans Syst Man Cybern 9(1):62–66
25. Jundale TA, Hegadi RS (2015) Skew detection and correction of Devanagari script using Hough transform. Procedia Comput Sci 45:305–311
26. Suzuki S, Abe K (1985) Topological structural analysis of digitized binary images by border following. Comput Vis Graph Image Process 29(3):396