Segmentation and recognition of characters on Tulu palm leaf manuscripts

P.J. Antony

Department of Computer Science,
AJ Institute of Engineering and Technology,
Mangalore, VTU, India
Email: antonypjhon@gmail.com

C.K. Savitha*

Department of Computer Science,
KVG College of Engineering,
Sullia, VTU, India
Email: cksavithaharish@gmail.com
*Corresponding author

Abstract: This paper proposes an efficient method for segmentation and recognition of handwritten characters from Tulu palm leaf manuscript images. The proposed method uses an automated tool with a combination of thresholding and edge detection technique to binarise the image. Further projection profile with connected component analysis is used to line and character segmentation. Deep convolution neural network (DCNN) model used here to extract features and recognise segmented Tulu characters efficiently with a recognition rate of 79.92%. The results are verified using benchmark dataset, the AMADI_LontarSet to generalise our model to handwritten character recognition task. The results showed that our method outperforms from the existing state of art models.

Keywords: handwritten character recognition; palm leaf; segmentation; deep convolution neural network; DCNN; Tulu.

Reference to this paper should be made as follows: Antony, P.J. and Savitha, C.K. (2019) 'Segmentation and recognition of characters on Tulu palm leaf manuscripts', *Int. J. Computational Vision and Robotics*, Vol. 9, No. 5, pp.438–457.

Biographical notes: P.J. Antony is serving as a Professor in the Department of Computer Science and Engineering at the AJ Institute of Engineering and Technology, Mangalore, under VTU Belagaum, Karnataka, India. He was received his PhD for Natural Language Processing in January 2013 from the Amrita Vishwa Peetham University, Coimbatore, Tamilnadu. His excellent contributions exhibited through publications in several national and international journals. His areas of interest are in natural language processing, digital image processing, machine learning and data mining.

C.K. Savitha working as an Associate Professor in the Department of Computer Science and Engineering at the KVG College of Engineering Sullia, affiliated to Visvesvaraya Technological University Belagaum, Karnataka, India. She obtained her UG in Electrical and Electronics Engineering and PG in Computer

Science and Engineering from the VTU Belagaum, Karnataka, India. She excels in digital image processing, machine learning, natural language processing and data mining. Her burning desire and passion for learning led to research in digital image processing and machine learning.

1 Introduction

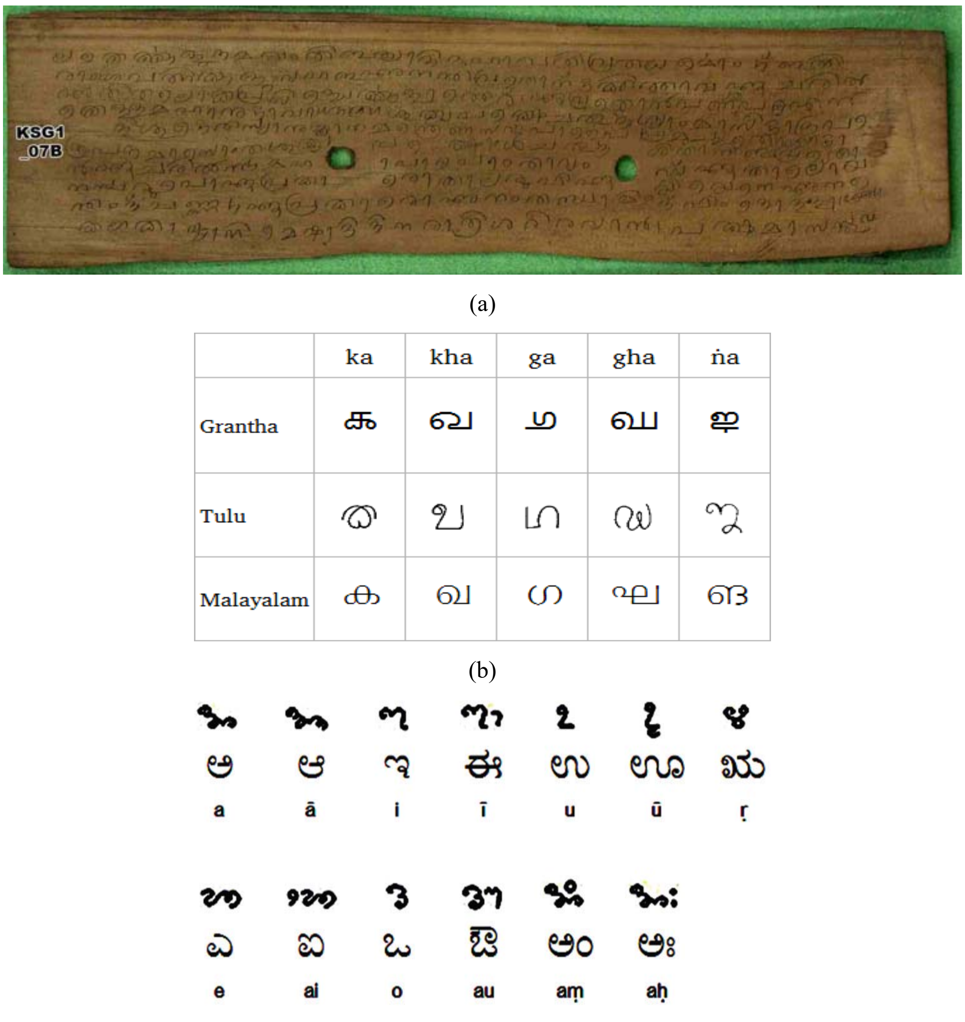
Ancient palm leaf manuscripts were one of the earliest sorts of written media and had been utilised in Southeast Asia to store early written information about subjects such as medicine, culture and astrology. Consequently, historical handwritten palm leaf manuscripts are crucial for people who like to study about historical documents. People can gain more experience from them. It is miles pretty common for such documents to be afflicted by degradation troubles including, the presence of smear, strains, the heritage of massive variations and uneven illumination, seepage of ink. Those files have no longer been well-preserved, still unprotected and damaged by the time. To hold these precious documents, handwritten optical character recognition is one of the first choices. It is very much important to convert them to the editable format in order that information in it may be used in today's era. The task is more complicated for Indian languages because of complexity involved in the shape and similarities of the characters (Inkeaw et al., 2015). The predominant problem is the big version within the writing forms of individuals at distinctive times and among exceptional individuals. Different difficulties include the similarity of some characters with every other, infinite variety of character shapes, broken and distorted characters.

Brahmi is the fundamental script used in India to represent inscriptions. The Tulu is one among the many evolved versions of Brahmi script (Poojary, 2012). The palm leaf manuscript in Figure 1(a) indicates the evolution of Tulu script since 12th century AD. The Tulu alphabet also referred to as Tigalari alphabet (Poojary, 2012), which was derived from the Grantha alphabet and resembles the Malayalam alphabet as shown in Figure 1(b).

Tulu is a language in the Southwest of Karnataka State and in part of Northern Kerala State in India (Antony and Savitha, 2016). The Tulu alphabet was used mainly for religious writings and is still used to some extent in parts of the Southern Region of Karnataka. Tulu speakers use the Kannada alphabet for official purposes since Kannada is the official language of Karnataka state. Kannada and Tulu scripts are Abugida written from left to right and both are used to write Sanskrit, Kannada, Tulu texts. The character set is almost identical to that of other Brahmi scripts and categorised as vowels, consonants and semi consonants. The evolution of the script has undergone modifications in the structure and shape of the alphabets due to reasons such as a method of writing, tools and materials used for writing of texts. Tulu, Kannada alphabets (Antony and Savitha, 2016) with ISO notation is as shown in Figure 1(c).

In the literature, it has been observed that no work is reported to recognise handwritten Tulu character from palm leaf manuscripts. We presented work on Tulu recogniser for modern documents (Antony and Savitha, 2016; Antony et al., 2016a). In this paper, we proposed a method for recognition of handwritten Tulu characters which are extracted from Tulu palm leaves. The preservation of these documents in Kannada editable form provides valuable insight into past history, cultures and civilisations. To reflect the generalisable ability for multiple languages, the proposed framework has been evaluated with Tulu palm leaf manuscript images and benchmarking dataset from AMADI_LontarSet, the first handwritten Balinese palm-leaf manuscript dataset (Burie et al., 2016). The work presented here involved a combination of thresholding with edge detection-based binarisation, projection profile with connected component (CC) analysis-based segmentation, Deep convolution neural network (DCNN)-based recognition (Liu et al., 2013). The related works are discussed in Section 2. The proposed system is explained in Section 3. Section 4 presents experimental results. Concluding remarks are given in Section 5.

Figure 1 (a) Tulu palm leaf manuscript (b) Similarity between Grantha, Tulu, Malayalam alphabets (c) Tulu, Kannada alphabets and ISO notation (see online version for colours)



2 Related work

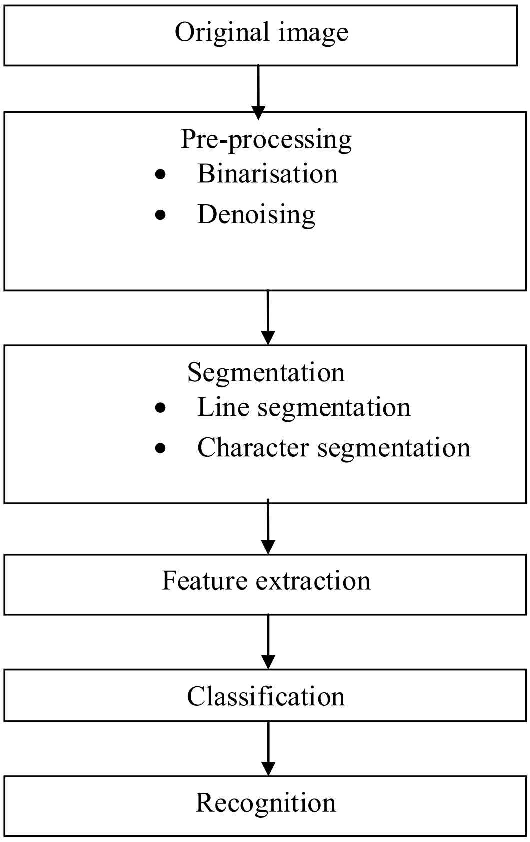
Document image analysis is one of the subfields of computer vision, which consists of stages such as image pre-processing, feature extraction and recognition (Fisher et al., 2013; Haralick and Shapiro, 1991). Pre-processing involves image enhancement in spatial or frequency domain as well as thresholding. Laplacian operator, adaptive bilateral filter and Gaussian filter-based hybrid spatial domain enhancement approach is described by Ranganatha and Holi (2015). Phattarachairawee and Ketcham (2017) presented contrast stretching, histogram-based enhancement on palm leaf manuscripts. Quality of image is improved using these enhancement techniques. Liu and Srihari (1997) presented Otsu's (1979) with texture feature-based thresholding algorithm for pre-processing of noisy documents with reasonable accuracy. Comparison of various approaches used for binarisation of document images is presented in Trier and Taxt (1995). Authors concluded that Yanowitz and Bruckstein were the best of the existing methods. Improved stroke edge-based binarisation algorithm for the palm leaf manuscripts is presented in Ge et al. (2017). After pre-processing, techniques such as projection profile analysis, improved Viterbi algorithm-based on hidden Markov model (HMM) (Ge et al., 2016), clustering algorithms (Valy et al., 2016) are used for line and character segmentation. To deal with segmentation of touching characters, fuzzy multi-factorial analysis is presented in Garain and Chaudhuri (2002).

Topographic (Lee and Kim, 1995), zoning (Deepa and Rajeswara Rao, 2014), wavelets features (Deepa and Rajeswara Rao, 2014) are used to represent the image with least amount of elements. Combination of features also used by Kesiman et al. (2016) to improve the recognition accuracy. Survey on offline recognition system for the handwritten script is presented in Jayadevan et al. (2011) and Arica and Yarman-Vural (2001). Recognition can be of template matched (Desai and Singh, 2016) or feature-based. Template matching techniques not suitable for the character with complex and similar structures. Lee and Kim (1999) described integrated segmentation and recognition system for handwritten numerals using cascade neural network model. This model works efficiently for a dataset with more connected characters. Mathematical morphology-based decision tree classifier for handwritten digit recognition system is presented (Mohammad and Husain, 2007). Recognition based on extracted features are explained using supervised learning models such as HMMs (Arica and Yarman-Vural, 2002; Prasad et al., 2008), convolution neural networks, support vector machines (SVM) (Elleuch et al., 2016), modified quadratic discriminant machines (MQDF) (Moni and Raju, 2011), artificial neural networks (ANN) (Marinai et al., 2005) etc. ANN requires large training time. Recent research attraction towards deep models due to its fast training time and improved denoising performance (Zhang et al., 2017). Recognition accuracy of 97% is achieved for handwritten Arabic characters (Boufenar and Batouche, 2017). From the literature, it has been observed that performance of the system depends on a suitable combination of pre-processing, feature extraction methods with classifier model.

3 Proposed system

The proposed methodology for degraded palm leaf manuscript image acquisition, pre-processing, segmentation, feature extraction, classification and recognition stages are illustrated in Figure 2 and described in following subsections.

Figure 2 Proposed system



3.1 Image acquisition

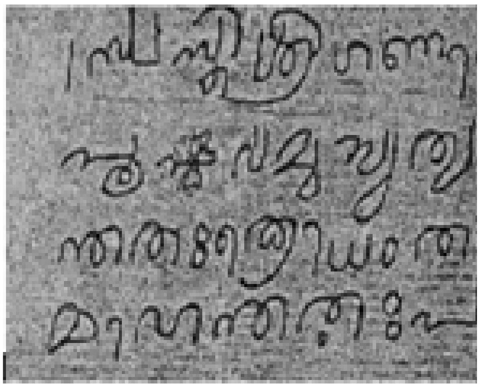
Image acquisition is a process of obtaining the data in digital format. It can be achieved by taking photograph or scanning the document. Sample image of acquired Tulu palm leaf manuscript is shown in Figure 3(a).

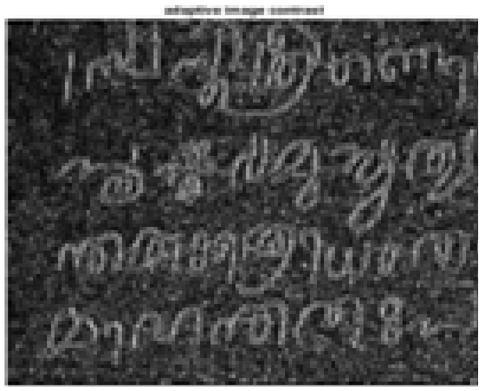
3.2 Image pre-processing

The main intention of pre-processing is to reduce variations in writing styles of different people. The procedures used in pre-processing steps are noise removal, size normalisation and grey scale conversion. It involves two stages as binarisation and denoising with enhancement technique.

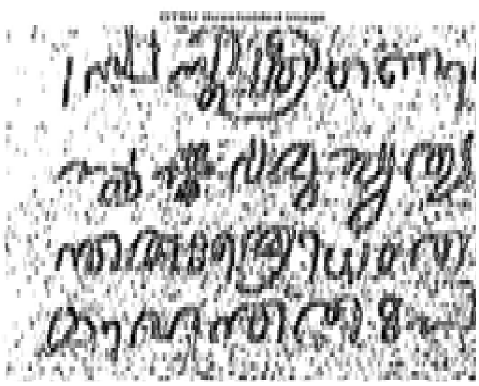
Figure 3 (a) Input image (b) Grey scale conversion of part of palm leaf (c) Adaptive contrast map (d) Otsu thresholding (e) Adaptive thresholding (f) Canny (g) Sobel edge detection (h) Otsu with canny (i) Otsu with sobel (j) Otsu with TV (k) Adapt with canny (l) Adapt with sobel (m) Adapt with TV

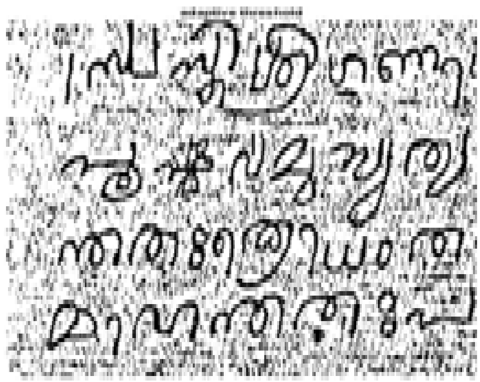
(a)





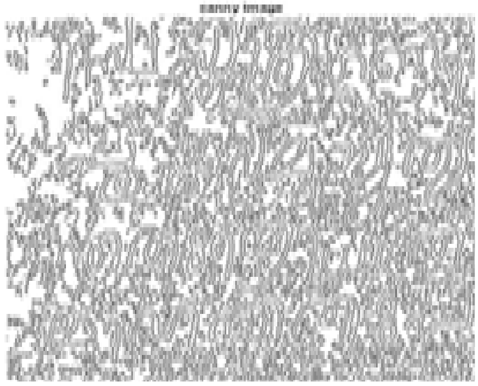
(b) (c)



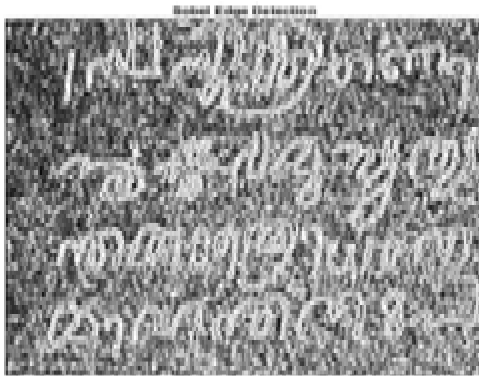


(d) (e)

Figure 3 (a) Input image (b) Grey scale conversion of part of palm leaf (c) Adaptive contrast map (d) Otsu thresholding (e) Adaptive thresholding (f) Canny (g) Sobel edge detection (h) Otsu with canny (i) Otsu with sobel (j) Otsu with TV (k) Adapt with canny (l) Adapt with sobel (m) Adapt with TV (continued)

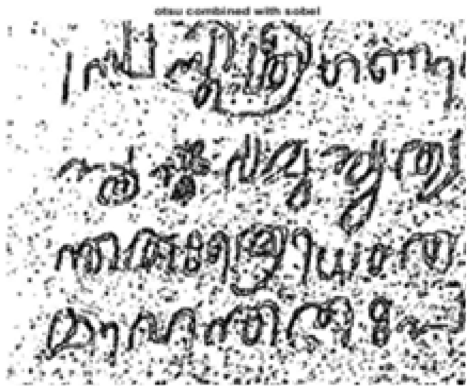


(f)



(g)

(h)

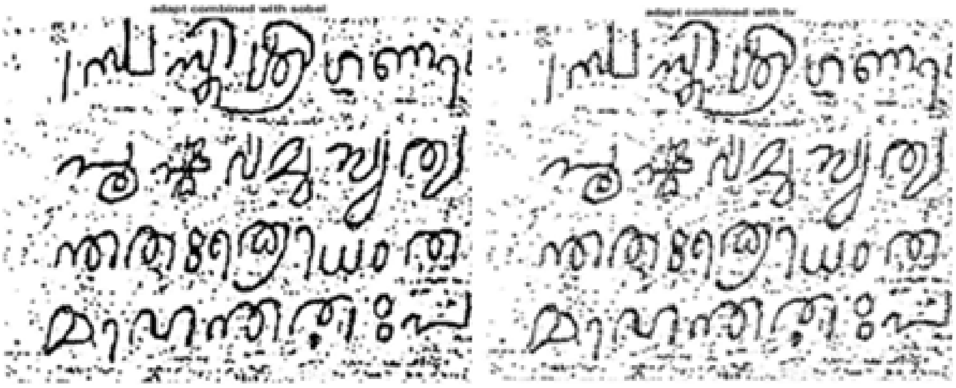


(i)

(j)

(k)

Figure 3 (a) Input image (b) Grey scale conversion of part of palm leaf (c) Adaptive contrast map (d) Otsu thresholding (e) Adaptive thresholding (f) Canny (g) Sobel edge detection (h) Otsu with canny (i) Otsu with sobel (j) Otsu with TV (k) Adapt with canny (l) Adapt with sobel (m) Adapt with TV (continued)



(l) (m)

3.2.1 Binarisation

Image binarisation is used for extracting objects from their background. Sometimes the input may be a colour image. Colour image has different pixel values. So, processing on colour image is a time-consuming task. Thus first the entire input colour image is converted into greyscale image as shown in Figure 3(b). After that, the greyscale image is converted into an image with only two values i.e., binary conversion. The need for converting to binary is to reduce the processing time for the Algorithms. Here we used binarisation technique which we presented in Antony et al. (2016b) for handwritten Tulu paper documents. Initially, adaptive contrast map of the palm image is constructed to handle bright pixels as shown in Figure 3(c). Then automatic selection of a better combination of thresholding with edge detection technique is selected by comparing peak signal to noise ratio (PSNR) and mean square error (MSE) values. Otsu's global thresholding method with canny, sobel, and total variation (TV) operators and adaptive thresholding technique with canny, sobel, and TV operators are applied to segment text stroke edge pixels from the noisy background. The resultant images of binarisation process such as Otsu thresholding, adaptive thresholding, canny, sobel edge detection, Otsu with canny, Otsu with sobel, Otsu with TV, adapt with canny, adapt with sobel, adapt with TV are shown in Figures 3(d)–3(m) respectively. The best combination is selected automatically by considering an image with high PSNR value as explained in Algorithm 1.

Algorithm 1

Require: The Palm leaf Image I

Ensure: The Binary Result B1 = Best_Psnr_Combined_Image

1 Compute gradient and local contrast of the image.

2 Construct adaptive contrast map Ac of the Image I

3 For each image Ac

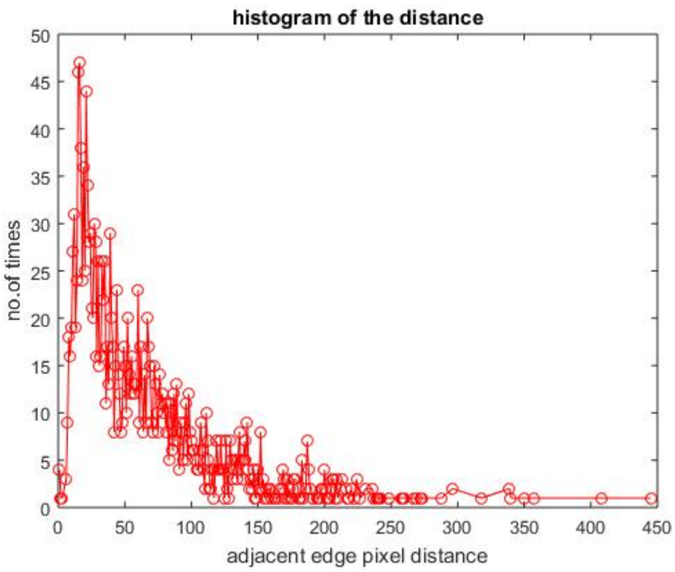
4 Apply Otsu thresholding Oth, Adaptive thresholding Ath, Canny, Total variation (TV), Sobel operator
5 if an image is threshold either Oth or Ath
6 Combine with Canny, Sobel or TV image
7 Compute [Psnr, pos] = [Oth_Canny, Oth_Sobel, Oth_TV, Ath_Canny, Ath_Sobel, Ath_TV]
8 Return Max [Psnr, pos]
9 else go to 4
10 end if
11 end for
12 Store binarised combined Image B1 with maximum Psnr.

After selection of suitable combination, the further processing is needed to extract only textual pixel $R(m, n)$ from combined thresholded with edge detected image. The text pixel $R(m,n)$ is extracted from text stroke edge pixels (Su et al., 2013) using equation (1) i.e.,

$$R(m, n) = \begin{cases} 1 & \text{if } I(m, n) \leq E\mu + \dfrac{E\sigma}{2} \\ 0 & \text{Otherwise} \end{cases} \tag{1}$$

where $E\mu$ and $E\sigma$ are mean and standard deviations of text stroke edge pixels intensity within neighbourhood window W. Window size is selected such a way that its size should be greater than stroke width of the image under study in order to carry text stroke edge pixels. Histogram of adjacent edge pixel distance is constructed as shown in Figure 4, to approximately estimate stroke width, hence to set window size.

Figure 4 Histogram of adjacent edge pixel distance (see online version for colours)

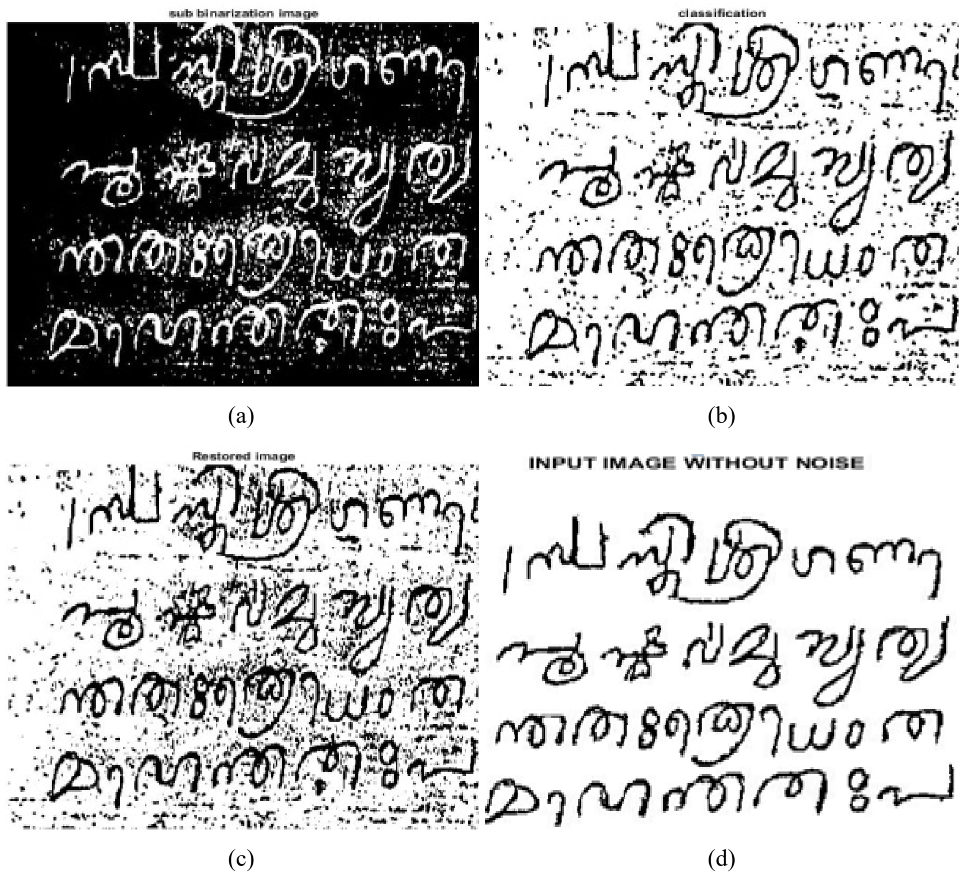


Sub-binarised image is shown in Figure 5(a). By incorporating CC analysis, an image is further improved as described in Algorithm 2. Initially, the CCs of text stroke edge pixels in sub-binarised images are extracted. Then for remaining edge pixels, its eight neighbourhoods are determined. Then pixel is labelled as text or background by comparing the intensity of neighbourhood pixel pairs. If intensity is lower than a predefined threshold, then it is assigned as text. Newly binarised image B3 is shown in Figure 5(b). Finally, additional noise along the text stroke boundaries are eliminated by applying logical operation on newly binarised image. Some morphological operations also applied to correct broken characters and to remove noise. Restored image is shown in Figure 5(c).

3.2.2 Denoising with image enhancement

The noise in the acquired data will reduce the system performance. To enhance the quality of the image, spatial filtering techniques such as mask can be applied. To remove the salt and pepper noise, the median filter is used. Image shown in Figure 5(c) which is an output from image binarisation may contain various noises e.g., small noises, leak noises and leaves streaks noise.

Figure 5 (a) Sub-binarised image (b) Newly binarised image B3 (c) Restored image (d) Enhanced binary image



Algorithm 2

Input: The Palm leaf Image I, Initial Binarised image B1 and Sub binarised image B2, threshold value

Ensure: The Final Binary Result B3

```
Obtain connect components of the text stroke edge pixels in B2 by removing other
individual pixels.
for remaining edge pixels (m, n)
find out its eight neighbourhood pairs
if The pixels in the adjacent pairs belong to the same category, either text or background
class then
if the intensity of pixel <threshold, Assign the pixel to text
else Assign pixel to background
end if
end for
Output the new binary result to B3.
```

Small noise is removed by CC labelling. Let P be the minimum number of pixels used to represent the object. CC analysis technique detects and removes the object in which, number of pixels less than P pixel. This operation is known as noise removal or area opening. For 2D, the default connectivity is 8. Connectivity value is 26 for 3D, and for higher dimensions, CONNDEF [NDIMS (BW), ‘maximal’] is used. Noise removal removes all CCs (objects) that have fewer than P pixels, producing another enhanced binary image as shown in Figure 5(d).

3.3 Segmentation

Segmentation process which is divided into two steps: line segmentation and character segmentation. For line segmentation step, projection profile analysis is a popular technique. Line detection uses the projection profiles include partitioning into vertical strips and horizontal run calculation as well as the calculation of the projection profiles of black pixels for each strip of the raster image and consider each obstructing handwritten CC by associating it to the text line above or below. Figure 6 shows line segmented image.

After the line segmentation have been found, the blocks of characters are segmented using the vertical projection profile of each text line. Properties of image regions such as the mean (μ) and standard deviation (σ) width of all the segmented blocks of characters in each document are calculated by plotting rectangular bounding box as shown in Figure 7.

The value is then used for the analysis of the width of the segmented blocks of characters. If the width of the segmented block is more than the criteria value, it will be separated again by using a CC labelling. Character segments are shown in Figure 8. The extracted character image is resized into standard size. Here we used the size of 28 × 28.

Figure 6 Line segmentation

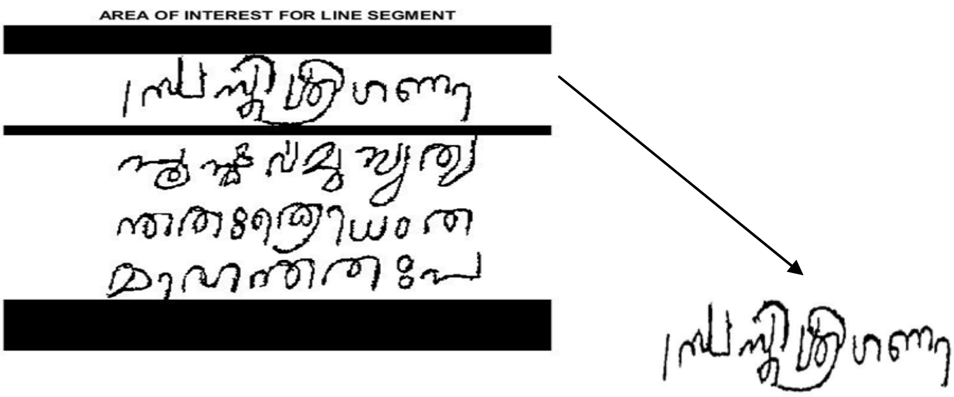


Figure 7 Bounding box (see online version for colours)

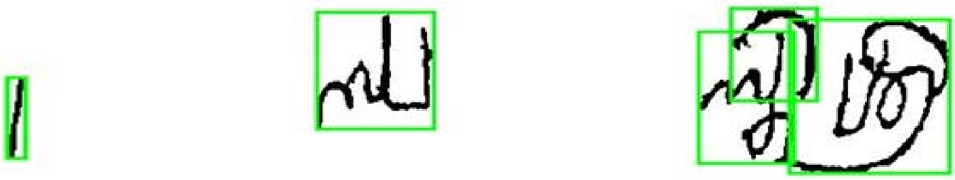


Figure 8 Characters segmented

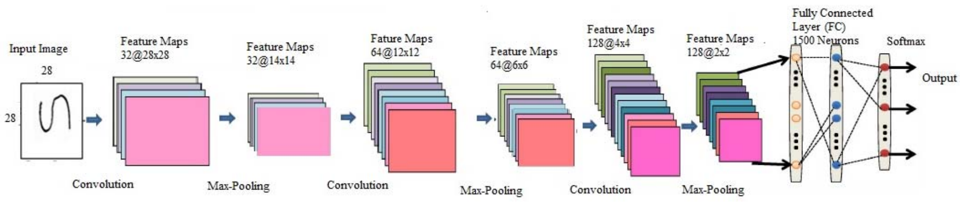
3.4 Feature extraction, classification and recognition

In this stage, the features of the characters which can be essential for classifying them at recognition stage are extracted. This is an important stage as its effective functioning improves the recognition rate and decreases the misclassification in reaching the good performance of handwritten recognition.

Deep learning is a subfield of machine learning, in which DCNN model includes input layer, feature detection layer, classification layer. Feature detection layer extracts topological structural information by passing segmented characters through convolution filters where pooling operation down samples the features which are used to learn the model. To achieve effective training, there is a need to map negative values to zero. To perform this, rectified linear unit (ReLU) of feature detection layer is used, which maintains only positive values. These operations are repeated over multiple nonlinear processing hidden layers. Shape variations of characters handled efficiently by detecting different features through each layer. Deep learning computational model reduces learning period of trainable feature vectors. Classification layer of DCNN includes fully connected layer which outputs vector with dimensions equal to the number of classes. Soft max function of final layer provides classified output. DCCNs architecture is adopted here for recognition purpose as shown in Figure 9. Here, the operations are organised into a multilayered feed forward network with alternating convolution, max-pooling layers followed by fully connected layer.

DCNN architecture is given with depth D, for the first layer, 32 filters of size $3 \times 3 \times c$ are used to generate 32 feature maps, and ReLUs (max(0, •)) are then utilised for nonlinearity. Max pooling layer down samples feature maps to size ~D-1, i.e., $2 \times 2 \times c$. For 2nd layer, 64 filters of size $3 \times 3 \times c$ are used with max pooling reduces size to $2 \times 2 \times c$. For the last layer, convolution filters of 128 with $3 \times 3 \times c$ and max pooling produces 128 feature maps with size $2 \times 2 \times c$, where $c = 1$ for binary and three for the colour image. Finally fully connected layer with 1,500 neurons outputs the features as well as character class. To sum up, our DCNN model uses residual learning and by incorporating convolution with ReLU, DCNN can gradually reduce noise and increase training speed through the hidden layers. Soft max activation function is used in the output layer.

Figure 9 DCNN architecture (see online version for colours)



4 Experimental results

4.1 Experimental setting

Our experiments are made on arbitrarily selected 25 Tulu palm leaf manuscript images from episodes of Anathavathara, Bhagavatha, Ramayana, Durga Stuthi which are collected from National Trust for Computation and Archival of Oriental Media, Udupi as well as from Mr. Ullal, Tulu philologist, Namma Tulunaadu Trust. To create binarised images from those digitised images, as an initial step, there is a need to extract textual content from the noisy background. We applied automated technique with a different combination of thresholding (such as Otsu, adaptive) and edge detection techniques (such as canny, sobel, TV) (Antony et al., 2016b) which we used in our previous work for historical paper documents. To evaluate the effectiveness of binarisation process there is a need of ground truth image. As there is no availability of ground truth image for our Tulu dataset, we used AMADI_LONTARSET, i.e., benchmark dataset to evaluate and measure the performance of our algorithm to binarise palm leaf manuscripts. Binarised images ground truth dataset for the AMADI_LONTARSET is shown in Figure 10 which includes 50 original images, 50 ground truth binary images version 1 (from the 1st ground truther) and 50 ground truth binary images version 2 (from the 2nd ground truther). For the testing subset, 50 original images (different from the training subset) have been provided (Burie et al., 2016). Results are evaluated based on PSNR values as shown in Table 1.

Results from Table 1, shows that proposed method gives best binarisation result with PSNR value of 53.3977 for baseline_dataset. The same metric is used for performance evaluation of binarisation of Tulu palm leaf images. Results of PSNR, MSE, time of execution for a combination of thresholding with edge detection techniques of Tulu palm leaf image selected from Anathavathara episode is shown in Table 2.

Figure 10 Benchmark dataset AMADI_LONTARSET (see online version for colours)

Table 1 Evaluation results of approaches presented in Burie et al. (2016) and our proposed work for baseline_dataset

No.	*Group*	*PSNR for 1st GT binary*	*PSNR for 2nd GT binary*
1	Yunxue Shao et al.	31.37	31.34
2	Chris Tensmeyer et al.	33.39	33.32
3	Su Bolan et al.	26.92	26.83
4	Dr. Deepak Kumar	29.98	29.92
5	Proposed method	54.3997	54.31

Table 2 PSNR, MSE, execution time for combination of thresholding with edge detection techniques

	Otsu_canny	*Otsu_sobel*	*Otsu_TV*	*Adapt_canny*	*Adapt_sobel*	*Adapt_TV*
PSNR	53.488	53.7581	53.7676	53.6462	53.6511	53.6514
MSE	0.2743	0.2737	0.2731	0.2808	0.2805	0.2805
TIME in ms	26.218	28.1582	27.8531	27.9898	28.6262	27.9211

Post processed image shown in Figure 11 selects Otsu_TV combination to give best binarisation result with highest PSNR value of 53.7676 and least MSE value of 0.2731 with reasonable execution time. Resultant isolated Tulu characters with horizontal projection profile analysis to line segmentation and vertical projection profile with CC analysis to character segmentation are shown in Figure 12.

Figure 11 Tulu palm leaf image with its binarised result (see online version for colours)

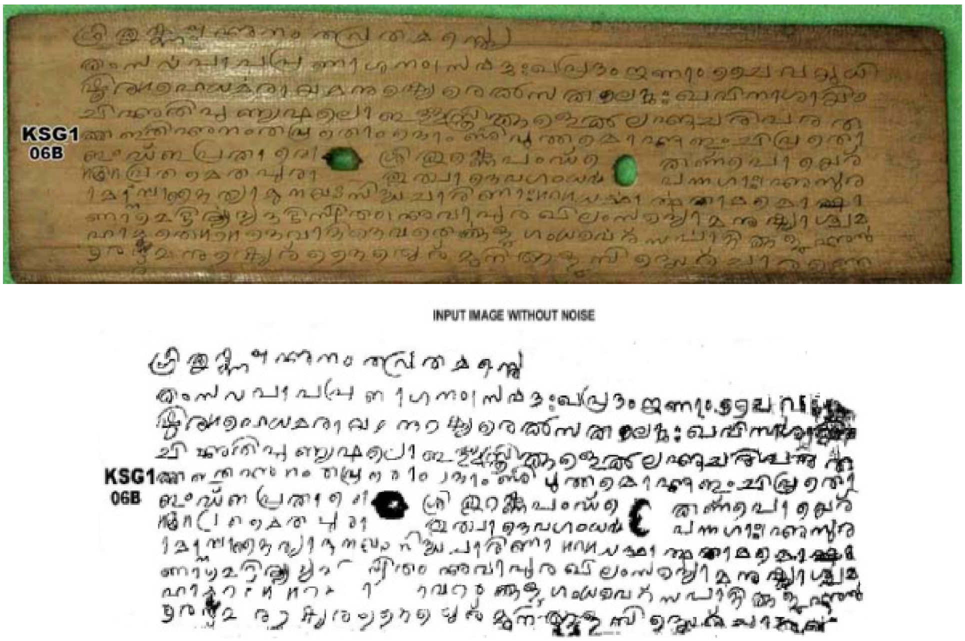
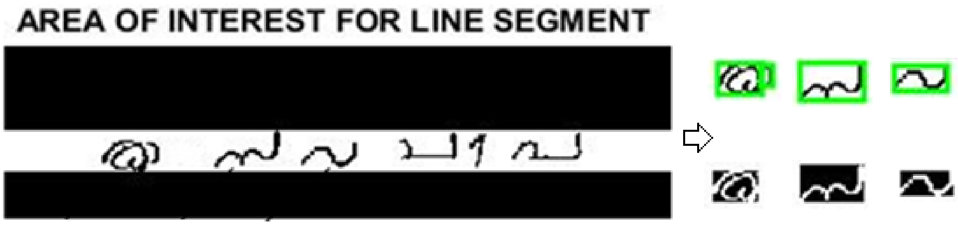


Figure 12 Isolated Tulu characters (see online version for colours)



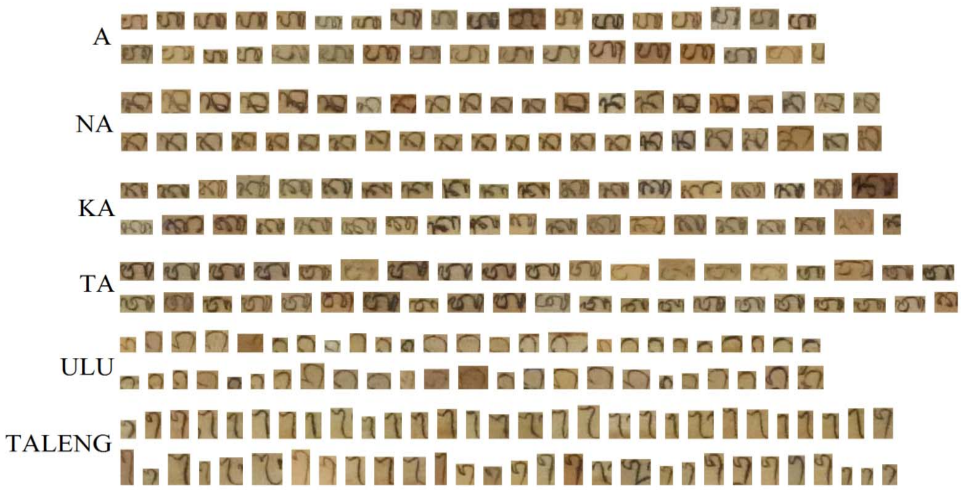
In order to recognise isolated Tulu characters, machine learning model called DCNN model is used. To design proper model to recognise handwritten characters, variety of handwritten samples are needed. So that we can adjust a number of neurons in various layers to maximise recognition efficiency. To evaluate the effectiveness of our deep model, we used deep model presented in Boufenar and Batouche (2017) by testing with AMADI_LontarSet, which was included with the alphabet and numeral of Balinese script. It is composed of ±100 character classes including consonants, vowels, diacritics, and some other special compound characters. Even though a 20 × 20 pixel rectangular image is sufficient to show details needed for better recognition (Cireşan and Meier, 2015), we rescaled all isolated images to 28 × 28 pixel images to allow all kinds of deformations. Identical pre-processing procedures reduce error rates, so we applied same pre-processing steps for training and testing samples. We got improved recognition speed by reducing the number of neurons in the fully connected layer from 1,500 to 500. Recognition results of AMADI_LontarSet with various approaches are presented in Table 3.

By selecting network architecture of 28x28-32C3x3-MP2x2-64C3x3-MP2x2-128C3 x3-MP2x2-500FC-133N where C represents convolution layer with filter size 3 × 3, MP represents max pooling layer with 2 × 2 pooling size, FC represents fully connected layer with 500 neurons and N represents number of neurons to represent number of character classes in output layer, we achieved recognition rate of 77.97% for AMADI_LontarSet dataset. Samples of character level annotated image is shown in Figure 13.

Table 3 Recognition rate of characters from AMADI_LontarSet with various methods

No.	*Group*	*Architecture*	*Recognition rate [%]*	*Speed [ms/character]*
1	Cristinel Codrescu et al.	Finite impulse response multilayer perceptron (FIRMLP)	77.70	Not mentioned
2	Su Bolan et al.	Random forest with CNN model	77.83	-
3	Proposed method	28x28-32C3x3-MP2x2-64C3x3-MP2x2-128C3x3-MP2x2-1500FC-133N	78.21	3.05
4	Proposed method	28x28-32C3x3-MP2x2-64C3x3-MP2x2-128C3x3-MP2x2-500FC-133N	77.97	2.16

Figure 13 Samples of character level annotated image (see online version for colours)

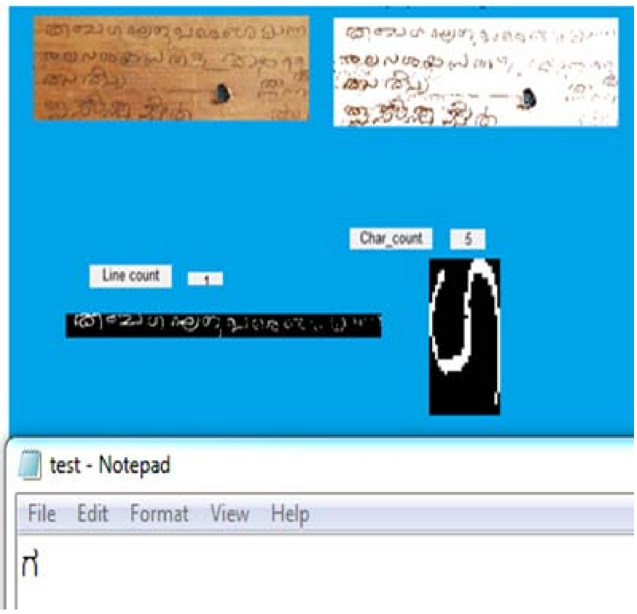


4.2 Training and testing Tulu data

We have used our own database to classify and recognise segmented characters from Tulu palm leaf manuscripts by using finalised DCNN architecture with 500 neurons in FC layer. Tulu character set includes more than 500 characters including vowels, consonants, vowel diacritics, compound characters. As our initial work we used a database containing 48 character classes i.e., 13 vowels and 35 consonants with 150 shapes per class. Due to difficulty in handling segmented compound characters as shown in Figure 8, we used only simple isolated characters. We faced one more difficulty in training data is that, character shapes collected from palm leaves for each class is not balanced. Even though it may effect for performance but cannot be avoided because it is not possible to get the same number of samples for all ancient characters (Kesiman et al., 2016). Figure 14 shows sample database for Tulu letter ‘ga’. The first row shows characters selected from palm leaf manuscripts. The second row shows characters from various paper documents as well handwritten samples of students from schools around coastal Karnataka. Figure 15 shows a representation of the recognised character in Kannnada Unicode. The learning rate setup is used here is eta start 0.001; eta factor 0.95; eta stop 0.000003. Training is fast at 162s/epoch for 112 epochs due to small network size. Recognition rate of 79.92% is achieved due to the variety of character sample from palm leaf as well as paper documents. Moreover, some of the binary images (Figure 15) are unclear and they have a major effect on the accuracy of character recognition. There may be little reduction in accuracy as we increase the number of character classes. We may get an improved result if we use the same model for the recognition of character only from modern documents. However current result shows improvement in the accuracy of state of the art methodologies used for palm leaf character recognition. The proposed model can be applied in a wide variety of application domains.

Figure 14 Training samples of Tulu letter ‘ga’ (see online version for colours)

Figure 15 Recognition of Tulu letter ‘ga’ and representation using Kannada Unicode (see online version for colours)



5 Conclusions

In this paper, a mixture of thresholding with edge detection-based automated technique is proposed for binarisation. Here we showed the automated selection of binarisation technique with different combinations which includes Otsu with canny, sobel, TV and adaptive with canny, sobel, TV resulted with the excellent binary image with PSNR value of 53.76. The experimental results on the AMADI_LontarSet showed that it could achieve 20% improvement over the state-of-the-art techniques. The binarised image is further post processed to lessen background noise in addition to correct damaged characters. Later, the projection profile with CC analysis is used to segment characters from degraded palm leaf manuscripts. Finally, DCNN architecture is utilised for character classification and recognition, in which residual learning is adopted to boost the training process as well as to reduce noise. Unlike conventional machine learning models such as ANN, which require massive training time, our DCNN model has the ability to reinforce overall performance with affordable training time. These are added essence to improve the character recognition rate. Experimental results demonstrated that the proposed technique produces recognition rate of 79.92% with speed of 2.16ms/character for recognition of isolated handwritten Tulu palm leaf characters. In future, we will investigate integrated segmentation and recognition models for coping with all sort of Tulu characters.

References

Antony, P.J. and Savitha, C.K. (2016) 'A framework for recognition of handwritten South Dravidian Tulu script', in *Conference on Advances in Signal Processing (CASP)*, IEEE, pp.7–12.

Antony, P.J., Savitha, C.K. and Ujwal, U.J. (2016a) 'Haar features based handwritten character recognition system for Tulu script', in *IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, IEEE, pp.65–68.

Antony, P.J., Savitha, C.K. and Ujwal, U.J. (2016b) 'Efficient binarization technique for handwritten archive of South Dravidian Tulu script', in *International Conference on Emerging Research in Computing, Information, Communication and Applications*, Springer, Singapore, pp.651–666.

Arica, N. and Yarman-Vural, F.T. (2001) 'An overview of character recognition focused on off-line handwriting', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 31, No. 2, pp.216–233.

Arica, N. and Yarman-Vural, F.T. (2002) 'Optical character recognition for cursive handwriting', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 6, pp.801–813.

Boufenar, C. and Batouche, M. (2017) 'Investigation on deep learning for off-line handwritten Arabic character recognition using Theano research platform', in *2017 Intelligent Systems and Computer Vision (ISCV)*, IEEE, pp.1–6.

Burie, J-C., Coustaty, M., Hadi, S., Kesiman, M.W.A., Ogier, J-M., Paulus, E., Sok, K., Sunarya, I.M.G. and Valy, D. (2016) 'ICFHR2016 competition on the analysis of handwritten text in images of Balinese palm leaf manuscripts', in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, IEEE, pp.596–601.

Cireşan, D. and Meier, U. (2015) 'Multi-column deep neural networks for offline handwritten Chinese character classification', in *2015 International Joint Conference on Neural Networks (IJCNN)*, IEEE, pp.1–6.

Deepa, R.N.A. and Rajeswara Rao, R. (2014) 'Feature extraction techniques for recognition of Malayalam handwritten characters: review', *International Journal of Advanced Trends in Computer Science and Engineering*, Vol. 3, No. 1, pp.481–485.

Desai, S. and Singh, A.G. (2016) *Optical Character Recognition using Template Matching and Back Propagation Algorithm*, PhD diss..

Elleuch, M., Maalej, R. and Kherallah, M. (2016) 'A new design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition', *Procedia Computer Science*, Vol. 80, pp.1712–1723.

Fisher, R.B., Breckon, T.P., Dawson-Howe, K., Fitzgibbon, A., Robertson, C., Trucco, E. and Williams, C.K.I. (2013) *Dictionary of Computer Vision and Image Processing*, John Wiley & Sons, Hoboken.

Garain, U. and Chaudhuri (B.B.) 'Segmentation of touching characters in printed Devnagari and Bangla scripts using fuzzy multifactorial analysis', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 32, No. 4, pp.449–459.

Ge, P., Yu, P., Li, H. and He, L. (2016) 'Text line segmentation using Viterbi algorithm for the palm leaf manuscripts of Dai', in *2016 International Conference on Audio, Language and Image Processing (ICALIP)*, IEEE, pp.336–340.

Ge, P., Yu, P., Li, H. and Li, H. (2017) 'Stroke edge based binarization algorithm for the palm leaf manuscripts', in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, IEEE, pp.778–782.

Haralick, R.M. and Shapiro, L.G. (1991) 'Glossary of computer vision terms', *Pattern Recognition*, Vol. 24, No. 1, pp.69–93.

Inkeaw, P., Chueaphun, C., Chaijaruwanich, J., Klomsae, A. and Marukatat, S. (2015) 'Lanna dharma handwritten character recognition on palm leaves manuscript based on wavelet transform', in *2015 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, IEEE, pp.253–258.

Jayadevan, R., Kolhe, S.R., Patil, P.M. and Pal, U. (2011) 'Offline recognition of Devanagari script: a survey', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 41, No. 6, pp.782–796.

Kesiman, M.W.A., Prum, S., Burie, J-C. and Ogier, J-M. (2016) 'Study on feature extraction methods for character recognition of Balinese script on palm leaf manuscript images', in *2016 23rd International Conference on Pattern Recognition (ICPR)*, IEEE, pp.4017–4022.

Lee, S-W. and Kim, S-Y. (1999) 'Integrated segmentation and recognition of handwritten numerals with cascade neural network', *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 29, No. 2 pp.285–290.

Lee, S-W. and Kim, Y.J. (1995) 'Direct extraction of topographic features for gray scale character recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 7, pp.724–728.

Liu, C., Liu, J., Yu, F., Huang, Y. and Chen, J. (2013) 'Handwritten character recognition with sequential convolutional neural network', in *2013 International Conference on Machine Learning and Cybernetics (ICMLC)*, IEEE, Vol. 1, pp.291–296.

Liu, Y. and Srihari, S.N. (1997) 'Document image binarization based on texture features', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 5,pp.540–544.

Marinai, S., Gori, M. and Soda, G. (2005) 'Artificial neural networks for document analysis and recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 1, pp.23–35.

Mohammad, F. and Husain, S.A. (2007) 'Character recognition using mathematical morphology', in *International Conference on Electrical Engineering*, ICEE'07, IEEE, pp.1–5.

Moni, B.S. and Raju, G. (2011) 'Modified quadratic classifier for handwritten Malayalam character recognition using run length count', in *2011 International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT)*, IEEE, pp.600–604.

Otsu, N. (1979) 'A threshold selection method from gray-level histograms', *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1,pp.62–66.

Phattarachairawee, S. and Ketcham, M. (2017) 'An algorithm image enhancement for segmentation palm-leaf manuscript', in *International Conference on Digital Arts, Media and Technology (ICDAMT)*, IEEE, pp.378–382.

Poojary, J.D. (2012) 'Tulu Lipi Terile', *A Book on Tulu Script*, Mumbai.

Prasad, R., Saleem, S., Kamali, M., Meermeier, R. and Natarajan, P. (2008) 'Improvements in hidden Markov model based Arabic OCR', in *19th International Conference on Pattern Recognition*, ICPR 2008, IEEE, pp.1–4.

Ranganatha, D. and Holi, G. (2015) 'Hybrid binarization technique for degraded document images', in *2015 IEEE International Advance Computing Conference (IACC)*, IEEE, pp.893–898.

Su, B., Lu, S. and Tan, C.L. (2013) 'Robust document image binarization technique for degraded document images', *IEEE Transactions on Image Processing*, Vol. 22, No. 4, pp.1408–1417.

Trier, O.D. and Taxt, T. (1995) 'Evaluation of binarization methods for document images', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 3,pp.312–315.

Valy, D., Verleysen, M. and Sok, K. (2016) 'Line segmentation approach for ancient palm leaf manuscripts using competitive learning algorithm', in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, IEEE, pp.108–113.

Zhang, K., Zuo, W., Chen, Y., Meng, D. and Zhang, L. (2017) 'Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising', *IEEE Transactions on Image Processing*.