DESIDOC Journal of Library & Information Technology, Vol. 42, No. 4, July 2022, pp. 227-233, DOI : 10.14429/djlit.42.4.17742

Optical Character Recognition for Printed Tamizhi Documents using Deep Neural Networks

Monisha Munivel* and V.S. Felix Enigo

Department of Computer Science and Engineering, Sri SivaSubramaniya Nadar College of Engineering, Chennai, India
**E-mail: moni.munivel@outlook.com*

ABSTRACT

Tamizhi (Tamil-Brahmi) script is one of the oldest scripts in India from which most of the modern Indian scripts are evolved. The ancient historical documents are generally preserved as digitised texts using Optical Character Recognition (OCR) technique. But the development of OCR for Tamizhi documents is highly challenging as many characters have similar shapes and structures with very small variations. In specific, for Tamizhi script it is very difficult to build an OCR as many characters are combined characters. This can be a single character formed by a single vowel/consonant or compound characters formed by combining vowels and consonants. This paper deals with the development of Tamizhi OCR for printed Tamizhi documents which is anticipated to perform efficiently irrespective of poor quality, noises and various input formats of Tamizhi documents. This is a preliminary study towards developing an OCR for handwritten Tamizhi inscription images that recognises text captured from onsite inscriptions. The developed Tamizhi OCR for printed text can produce an accuracy of about 91.12 per cent.

Keywords: Tamizhi script; Tesseract OCR; Tamizhi documents; CNN-RNN-CTC networks

1. INTRODUCTION

OCR is a technology that recognises the text in the image and converts them into machine editable text. OCR supports almost all international languages character recognition except some ancient scripts that were mostly developed before BCE. A script generally refers to a set of graphical representations for the verbal form of communication.

Tamizhi script is one of the oldest and most important writing systems in the world for which full-fledged OCR is not developed so far. Most of the South Indian languages have their origin evolved from this script. The oldest Tamizhi inscriptions are seen in historical rock-cut edicts of Ashoka. It is also found in the caves, Jaina beds, and potteries of 3rd BCE to 6th CE which were excavated from many archaeological sites. It continued to develop over centuries and evolved as different scripts in different regions[1]. About 198 different modern scripts in Central and South Asia are said to have their origin from the Brahmi script.

While OCR generally can recognise printed characters with better accuracy for document images[2], Tamizhi script poses difficulty as most of the characters have very few variations[3]. Recognition of the Tamizhi script is needed because most of the valuable information about ancient days such as musical notations, types of donations, constructional details, etc. are written in Tamizhi scripts[1]. Also, the study of Tamizhi script helps to identify its relationship with the south Asian scripts[3].

Towards this initiative of automated text recognition, preliminary work was done by the authors Neha and Soo, for printed Brahmi vowels and consonants using the geometric method. It excludes 3 consonants (𑀨, 𑀫, 𑀮) and compound characters. The authors with the Matlab OCR for character recognition used 50 samples of 42 characters each[3].

Received : 01 December 2021, Revised : 30 March 2022
Accepted : 05 May 2022, Online published : 19 July 2022

1.1 Our Contributions

- To develop a complete OCR for Tamizhi documents from the scratch with a high character recognition rate
- To incorporate in Tesseract, an open-source OCR Engine for Tamizhi script recognition
- To use deep neural network with a large dataset for enhanced accuracy
- To develop a complete language data for Tamizhi script

The rest of the paper is organised as follows: Related work is discussed in section II. Section III explains the Tamizhi script and its characteristics. Section IV describes the system methodology. Results and analysis are discussed in section V. Finally, section VI deals with the conclusion and directions for future work.

2. RELATED WORK

There is ample work done in the development of OCR and its related processes for various scripts. Most of the research works focus on addressing the issues present in the steps involved in OCR to improve the efficiency of the recognition.

To recognise the Tamizhi script, the earlier heuristic binary method used the length of runs and the number of ones present in the image[1]. In this, each character is converted into a binary array of zeros and ones in which zero represents a blank and one represents a non-blank. Then, the binary array is transformed

into pairs of small strings. For all printed characters, strings are obtained and built into a dictionary. Test input characters are converted into the binary array before being fed into the tool for recognition. But the classification and prediction need manual intervention. Later methods such as the Hidden Markov Model (HMM) and Neural Networks classification and prediction of characters are performed automatically with better character recognition rates[4].

For generating synthetic text-line images dataset for the Greek polytonic script, OCRopus utility is used[5]. It uses many parameters such as blur, threshold, size, and skew which can be altered to produce synthetically generated text-line images that closely resemble the scanning process.

One dimensional bi-directional LSTM is used in recognition of the Ethiopian script. Text-line normalisation is applied as pre-processing step to the input text-line images for smoothing before it is trained using 1D LSTM[4]. In this, the centre normalisation method is applied to adjust the translation in the vertical axis of the characters in the script. It works perfectly irrespective of differences in the height of the image.

In the printed Latin scripts, character position with reference to the baseline is an important feature. Because most of the character pairs are similar and are distinguished mainly by the position and size with respect to the baseline. Hence, geometric normalisation[2] is found to be more reliable than text-line normalisation.

The next step in OCR is feature extraction which is a dimensionality reduction technique that is used to extract the important features of the image as a reduced feature vector. For extracting features in Tamizhi characters, zone-based methods and geometric methods were employed. In zone-based methods, each character of a Tamizhi script is divided into various zones for better classification. But, in the geometric method, the type of features was used for the classification of characters. Since Tamizhi inscriptions are short and precise, zone-based methods fail to capture details of the character. But geometric methods extract features in more detail by using the corner points, intersect points, ending points, circles, semi-circles, and bifurcation points[3].

Since many scripts are evolved from the early scripts, an algorithm that uses directional features to recognize evolved scripts was tested for Sinhala and Ethiopic scripts[6]. The steps in OCR are implemented at the abstract level which enables the same algorithms to be extended for different scripts with little modifications. The word-level accuracy was further improved using HMM which used a confusion matrix and a state transition matrix. The confusion matrix is created to spot the intra and inter-group confusing characters. Whereas, the state transition matrix built using Lexicon computes the probability of each character in the script for the current word.

For recognition of handwritten medieval Latin texts images, Deep Learning Recurrent Neural Network (RNN) learning approach and Connectionist Temporal Classification (CTC) approach was employed[2]. Though both the approaches performed well for recognition, the CTC approach outperforms the Seq2Seq approach for long words. It is because CTC predicts characters dynamically. But, Seq2Seq converts the whole image as a vector and classifies it to the predicted label.

3. TAMIZHI SCRIPT

Tamizhi script is one of the oldest scripts in India which is written in Tamil and Prakrit languages. Tamil is one of the oldest spoken languages whereas Prakrit is one of the earliest written languages. Between these languages, Tamil is one of the oldest living languages in the world with rich literature[7]. Evolved from the Ancient Tamizhi script, it is written from left to right[8]. It is a combination of alphabetical and syllabic systems. It has a relatively small character set of pure 12 vowels and 18 consonants. Additionally, it consists of 6 consonants known as Grantha letters which are borrowed from Prakrit. Apart from this, a character called Adytam, neither a consonant nor a vowel is used in Tamil Grammar[7]. Often, one or more consonants with vowels or a cluster of two consonants lead to a modified shaped character called vowel diacritic[8].

3.1 Characteristics of Tamizhi Scripts

When vowel and consonant are combined, depending upon the location of the vowel marker, different characters are formed. Such characters are called modified characters. Compound characters have a compound orthographic shape which is formed by altering the basic shape of the character by adding any one of the following to the consonant: a) vowel marker b) vowel-specific suffix or prefix and c) vowel-specific suffix and prefix[7]. Any character in Tamizhi is either a pure consonant or a combination of a syllable with a consonant and a vowel. The same consonant with a different vowel formed by drawing spare strokes attached to the character is called matras. Consonant clusters are called Nexus[9]. As most of the characters are formed by very little variation in shape, it poses a challenge in recognizing the characters by OCR, resulting in a low recognition rate[7].

4. METHODOLOGY

The proposed system accepts the printed Tamizhi documents as input to the OCR and recognises the text from the image. It is implemented in Tesseract[10] which is an open-source optical character recognition tool. This work uses the latest version of tesseract which supports deep neural network models. It is a command-line-based tool that has pre-built models and provisions to create models for new scripts and languages. Since Tamizhi script OCR is not provided by Tesseract, the tool must be trained for Tamizhi script to recognise the Tamizhi text.

A single-stage recognition method produces acceptable accuracies in small class problems. But, it fails to provide similar accuracy in large class problems. To overcome this issue, a multistage recognition system is used with fewer classes in each stage[8]. To build the complete OCR application for Tamizhi scripts the steps are as follows.

- Preparing training data
- Pre-processing the document image
- Performing recognition using Tesseract OCR
- Post-processing the generated output text

4.1 Preparing Training Data

In this process, the tool learns the features of characters from the training text image (which contains vowels, consonants,

and its variants of Brahmi script as a single image file) from each processing step and maps the features of a particular character to its corresponding ground truth. The final training data set includes a combination of all the vowels, consonants, numerals, consonants, and vowel modifiers, consonants and consonants modifiers, and compound characters with a vowel or consonant modifier. During the training preparation, the different sets of training data with image Dots per Inch (DPI), font types and sizes, types of the document image is used. All the above combinations use the image DPI of 300, font type Noto Sans Brahmi and font size of 24.

4.2 Pre-processing the Document Image

Pre-processing the document image helps to improve the chance of successful recognition. It involves a series of operations performed to enhance the image for character recognition[7]. In the pre-processing, the first step is the separation of the text from its background and the subsequent steps involve various operations over the extracted text to arrive at normalised individual characters suitable for character recognition. If the document image is not aligned properly during scan or photograph, it is necessary to tilt the image either clockwise or anticlockwise to certain degrees for proper orientation of the text with respect to the horizontal axis[6] as character segmentation and recognition requires text to be aligned properly.

Binarisation is the process of converting a grayscale image into a binary image. As binarisation affects the quality of character recognition significantly, careful selection of binarisation technique is needed based on the type of image used. With respect to the Tamizhi script, it has numerous small strokes. When binarisation is applied, these tiny strokes will be removed as their pixel value is comparatively less than the threshold. To retain these tiny strokes, dilation is applied before binarisation. Dilation is a morphological operation applied to the structuring element (boundaries) of the text where the pixels grow in size reducing the holes. It is based on the idea that the larger the text, the easier it will be for recognition by the OCR. Dilation results in short and wide or very tall and skinny characters. To bring this to equal size, the character aspect ratio is used. Aspect ratio is the ratio of the width of the character to the height of the character.

4.3 Performing Recognition using Tesseract OCR

The next step in building OCR is the recognition of text. The Tesseract training is performed using CNN-RNN-CTC deep neural network architecture. Tamizhi document images are given as training data for the network. The reason for using this architecture is, OCR is an image-based sequence recognition problem. The Convolutional Neural Network (CNN) is best suited for the image-based problem and Recurrent Neural Networks (RNN) is good at handling sequences. But RNN is limited in handling long-term dependencies which are highly required for text recognition. Hence, a variant of RNN, bidirectional Long Short Term Memory (LSTM) is used. LSTM is capable of handling long-term dependencies by remembering the contextual information of the whole sequence for each timestep. The output of LSTM is the probability of text present in the image. Further, Connectionist Temporal Classification (CTC) does the task of classification of sequences into given characters. It predicts the output by assigning the character with the highest probability to every input sequence[10].

Once the tool is trained to recognise the text, the next step is giving input or test data for recognition. Here, the input (printed Tamizhi document) is pre-processed for better recognition before it is fed into OCR. Tesseract supports Leptonica, the image processing package to optimise the images for efficient character recognition.

The next step is the character recognition process. Here, the normalised characters undergo a recognition process using a template matching algorithm or feature extraction depending upon the complexity of the character. Template matching is apt for typewritten characters. But feature extraction works even for new fonts. Template matching compares the image with the stored glyph pixel-by-pixel to recognise a character. Whereas, feature extraction decomposes the glyphs into striking features such as lines, closed loops, etc. of a character. Later, these glyph features are compared with image features using the nearest neighbourhood classifier to classify them into an appropriate class to recognise a character.

4.4 Post-processing the Generated Output Text

The final step in the character recognition process is post-processing. Character recognition may be constrained by the limited words in the lexicon. To improve this, linguistic databases and dictionary that contains the grammar of the script are used in the character segmentation step to improve the accuracy[11]. Further, the Levenshtein distance algorithm which computes the edit distance (number of changes needed to change one word to another) between two words is used as a post-processing algorithm to obtain results with good accuracy.

5. RESULTS AND DISCUSSION

5.1 Dataset

Ground truth data for both training and testing is essential for character recognition. Though many scanned documents are available to prepare the ground truth, a lot of work is required to create a realistic database that covers all the possibilities of Tamizhi scripts. Up to our knowledge now, there is no standard ground-truth dataset available for the Tamizhi script for OCR purposes. So, a Tamizhi database is developed which serves as a source for training and testing. The database contains 259 pages of text-line images and out of that 211 pages are used for training and 48 pages for testing. The training sample set is composed of 12 vowels, 18 consonants, and 216 consonantal vowels totalling 246 characters. The text files are converted into text-line images with the help of font files. The UTF-8 encoding method is used to generate the ground truth for each character and it has .otf (open type file) extension. The quantity and quality of the data influence the performance of the model.

5.2 Dataset Normalisation

Normalisation is a process that is used to adjust the transformation of the text-line images on the vertical axis.

Characters in the text-line images are centralised and rescaled by computing the mean absolute deviation of the smoothed image[4]. Text normalisation is the process of normalizing the text to a specific height. The text-line images in the dataset must be normalised to 36 pixels. The anti-aliasing technique is used to smoothen the jagged edges of the dataset by averaging the pixels in the boundary. Skew angles are determined and corrected.

5.3 Recognition Accuracy for Individual Characters

To precisely identify the characters which caused the errors, we performed a character recognition test by splitting the entire Brahmi character set into various small groups of vowels, consonants, numerals and conjunct characters as shown in Table 1.

Vowels: like 𑀅(a), 𑀆(aa), 𑀇(i), 𑀈(ii), 𑀉(u), 𑀊(uu), 𑀏 ,𑀐 ,𑀑 ,𑀒 ,𑀓 ,𑀔, 𑀕

Consonants: 𑀖, 𑀗, 𑀘, 𑀙 ,𑀚 ,𑀛 ,𑀜 ,𑀝 ,𑀞 ,𑀟 ,𑀠, 𑀡, 𑀢, 𑀣, 𑀤

Vargas*: 𑀥, 𑀦, 𑀧, 𑀨, 𑀩, 𑀪, 𑀫 ,𑀬 ,𑀭 ,𑀮, 𑀯, 𑀰, 𑀱, 𑀲, 𑀳, 𑀴

*Vargas is set of characters that are uttered with the breath and not the voice (e.g. Ka - consonant, Kha - Vargas)

Table 1. Tamizhi consonants

Consonants group number	Consonants group name	Group members
1	Guttals (Ka varga)	𑀖, 𑀥, 𑀦, 𑀧, 𑀗
2	Palatals (Ca varga)	𑀘, 𑀨, 𑀩, 𑀪, 𑀙
3	Cerebrals (ta varga)	𑀚, 𑀫 ,𑀬 ,𑀭 ,𑀛
4	Dentals (ta varga)	𑀜,𑀫, 𑀮, 𑀯, 𑀝
5	Labials, Semivowels and Sibilants	𑀧, 𑀨 ,𑀩 ,𑀪, 𑀰 ,𑀱, 𑀲 , 𑀳, 𑀴 ,𑀵, 𑀶, 𑀷, 𑀸, 𑀹, 𑀺

Numerals: 𑁒, 𑁓, 𑁔, 𑁕, 𑁖, 𑁗, 𑁘, 𑁙, 𑁚 ,𑁛, 𑁜, 𑁝, 𑁞, 𑁟, 𑁠, 𑁡, 𑁢, 𑁣 , 𑁤

Conjunct characters with the diacritic dot on the left or right side or both sides are often misclassified. E.g. 𑀶 and 𑀷. Also, characters with punctuation dots and punctuation hyphens are wrongly classified as they look similar. The recognition performance for various Tamizhi characters in various groups for 50 samples/characters is shown in Table 2.

5.4 Comparison of Method using Deep Neural Network (DNN) Vs Geometric Method

We compared the character recognition accuracy of our proposed Tamizhi OCR with that of the geometric based method of Neha Gautam et. al. The comparison results are shown in Fig. 1 and it is evident that the character recognition accuracy of Tamizhi OCR outperforms for all the groups compared to that of the geometric based method.

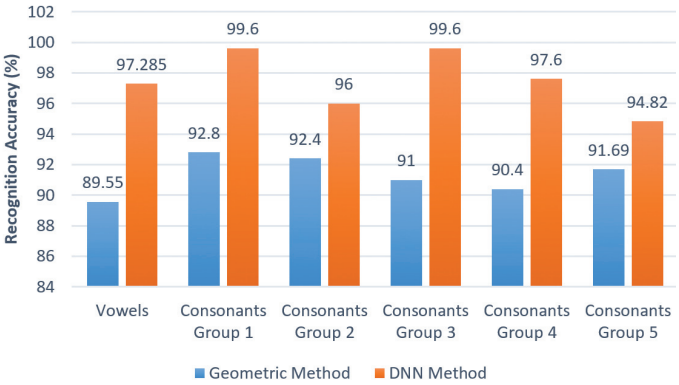

Figure 1. Geometric method Vs DNN method.

As mentioned in the literature, Neha Gautam et al. use geometric features such as circles, semicircles, corner points, ending points, intersect points and bifurcation points as the entities to learn the features of the character[3]. But, in our work, we have used LSTM training to train the network and it learns the character features automatically. Though DNN method produces an accuracy of about 97.29 per cent for vowels and 97.52 per cent for consonants in comparison with Geometric method, some characters are partially recognised it produces an accuracy of about 93.30 per cent for vowels and 94.90 per cent for consonants excluding some characters. Hence, our

Table 2. Recognition accuracy of Tamizhi OCR

Glyph groups	No. of characters in the group	Total No. of samples	Correctly recognised characters	Wrongly recognised characters	Error rate	Accuracy
Vowels	14	700	681	19	2.71	97.285
Consonants Group 1	5	250	249	1	0.40	99.60
Consonants Group 2	5	250	241	9	3.60	96.00
Consonants Group 3	5	250	249	1	0.40	99.60
Consonants Group 4	5	250	244	6	2.40	97.60
Consonants Group 5	17	850	806	44	5.17	94.82
Numerals	20	1000	946	54	5.40	94.60
Conjunct character	518	25900	22619	3281	12.67	87.33
Overall results		29450	26035	3415	4.09	95.85

Conjunct Characters: Conjunct characters are compound characters that are formed by combining vowels and consonants. E.g. the conjunct character for the 'ka' row are 𑀶, 𑀷, 𑀸, 𑀹, 𑀺 , 𑀻, 𑀼, 𑀽, 𑀾, 𑀿, 𑁀 and 𑁁.

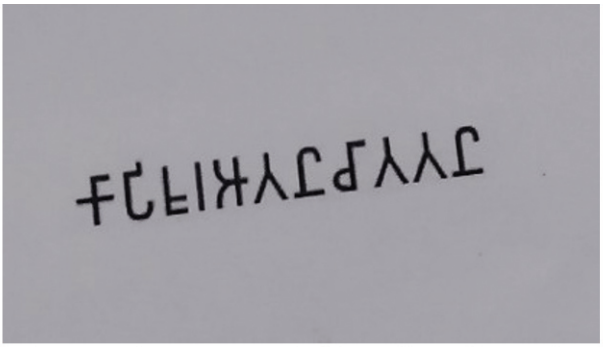

Figure 2. Input image.

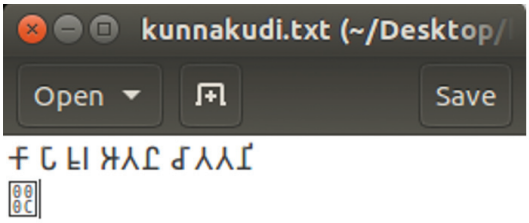

Figure 3. OCR text.

proposed DNN method is greatly better in terms of accuracy than the geometric based method.

5.5 Performance of OCR

The performance of OCR is improved by the processes specified in the system methodology. But the classification accuracy depends on both the scanning and the textual errors. Scanning errors are the errors induced during the process of scanning the images. Some of the scanning errors[12] are: digitisation error, paper quality, ink and dirt spattering, random distortions, quality of writing tools, etc., Textual errors are the errors that are part of the language. For example, it could be context-based or grammar-based.

Thus, the input image and its corresponding extracted OCR text are shown in Fig. 2 and Fig. 3. Best results for recognition are obtained only for the text with the same size and same font. But, we have generalised the approach to recognise different fonts and sizes.

The performance of the OCR was evaluated with the input text from Thirukkural and Tolkappiyam. In Thirukkural, Kural 1 to Kural 10 from the chapter Invocation was used to test the performance of OCR. For Thirukkural, an accuracy of 92.30 per cent is obtained for the printed Tamizhi script as given in Table 3. In Tolkaippiyam, the first 10 sutras from Chapter 1. Conventions of Phonology and Orthography were given as input text and for that, an accuracy of 92.06 per cent was obtained as given in Table 4.

We have tested the performance of OCR for different types of document images and the screenshot images taken using a

Table 3. Performance of OCR for Thirukkural

Kural No.	Total number of characters	Correctly classified characters	Misclassified characters	Error rate	Accuracy
1	26	22	4	0.1538	0.8461
2	30	26	4	0.1330	0.8666
3	27	24	3	0.1111	0.8889
4	31	30	1	0.0270	0.9677
5	33	30	3	0.0909	0.9090
6	32	31	1	0.0313	0.9688
7	36	34	2	0.0550	0.9444
8	36	35	1	0.2778	0.9722
9	32	30	2	0.0625	0.9375
10	28	26	2	0.0714	0.9285

Table 4. Performance of OCR for Tolkappiyam

Tholkappiyam Sutras No.	Total number of characters	Correctly classified	Misclassified	Error rate	Accuracy
1	52	49	3	0.0577	0.9245
2	43	38	5	0.1162	0.8837
3	41	36	5	0.1219	0.8780
4	37	34	3	0.0810	0.9189
5	18	17	1	0.0555	0.9444
6	32	30	2	0.0625	0.9375
7	31	30	1	0.0625	0.9375
8	25	23	2	0.0870	0.9200
9	24	22	2	0.0833	0.9166
10	18	17	1	0.0555	0.9444

Table 5. Performance of OCR for different Tamizhi document images

Types of Tamizhi document images	Number of images tested	Accuracy (%)
Old Scanned documents	58	88.54
Low quality documents	25	80.84
New Printed Documents	103	97.35
Screenshots – Font size 6 to 12	49	95.49
Screenshots – Font size 14 to 20	53	91.24

print screen and observed its performance. We found that the accuracy of the OCR depends on the quality of the input image and its resolution.

From Table 3-Table 5 the overall accuracy obtained for various testing sources is found to be 91.12 per cent. The loss in accuracy is mainly due to the misclassification of the complex characters [illegible], [illegible], [illegible] because of their similar and complex structures. Moreover, it occurs only in Tamizhi databases at very few places and not in any other historical sources. When the character is used in the same word as it is present in the linguistic database it is recognised perfectly. But it is misclassified if there is any word segmentation and the segmented word refers to a different context that is not present in the database and the training set.

6. CONCLUSION

A new OCR has been developed to recognise printed Tamizhi scripts. The input printed Tamizhi images collected from various sources are used for training and testing the model. Various pre-processing steps were implemented to develop the ground truth for the printed Tamizhi images. The CNN-LSTM networks were configured, trained and the language model for the Tamizhi script was developed. The model has been tested and the character error rate is negligible i.e. 0.474 per cent which is less than 1 per cent. It was found that this error is due to similar shapes of characters in the input images. But the overall accuracy of the OCR is found to be 91.12 per cent. Though the character error rate is very less, the fall in OCR accuracy is due to the 3 complex characters [illegible], [illegible], [illegible].

In the future, OCR accuracy can be improved by generating linguistics rules for the Tamizhi script. Further, OCR for Tamizhi script can be developed for inscription images.

REFERENCES

1. Siromoney, G.; Chandrasekaran, R. & Chandrasekaran, M. Machine recognition of Brahmi script. *IEEE Transact. Syst., Man, Cybernetics*, 1983, **7**(4), 648-54. doi: 10.1109/TSMC.1983.6313155
2. Shkarupa, Y.; Mencis, R. & Sabatelli, M. Offline handwriting recognition using LSTM recurrent neural networks. *In* 28th Benelux conference on artificial intelligence 10-11 November 2016, Amsterdam, The Netherlands. https://www.researchgate.net/publication/311672974_Offline_Handwriting_Recognition_Using_LSTM_Recurrent_Neural_Networks. (Accessed on 16 September 2021)
3. Gautam, N. & Chai, S.S. Optical character recognition for Brahmi script using geometric method. *J. Telecommun., Electron. Comput. Eng. (JTEC)*, 2017, **9**(3-11), 131-136. https://jtec.utem.edu.my/jtec/article/view/3197. (Accessed on 24 September 2021)
4. Addis, D.; Liu, C.M. & Ta, V.D. Printed ethiopic script recognition by using lstm networks. *In* 2018 International Conference on System Science and Engineering (ICSSE) 2018, 28-30 June 2018, IEEE pp. 1-6. doi: 10.1109/ICSSE.2018.8519972.
5. Simistira, F.; Ul-Hassan, A.; Papavassiliou, V.; Gatos, B.; Katsouros, V. & Liwicki, M. Recognition of historical greek polytonic scripts using lstm networks. *In* 2015 13th International Conference on Document Analysis and Recognition (ICDAR), 23-26 August 2015, IEEE (pp. 766-770). doi: 10.1109/ICDAR.2015.7333865.
6. Premaratne, L.; Assabie, Y. & Bigun, J. Recognition of modification-based scripts using direction tensors. *In* 4th Indian Conference on Computer Vision, Graphics and Image, 16-18 December 2004, Kolkata, India 2004 (pp. 587-592). https://www.cse.iitb.ac.in/~sharat/icvgip.org/icvgip2004/proceedings/ip3.8_276.pdf (Accessed on 16 September 2021)
7. Punitharaja, K. & Elango, P. Tamil handwritten character recognition: Progress and challenges. *Int. J. Control Theory Appl.,* 2016, **9**(3), 143-151. https://serialsjournals.com/abstract/18195_16-punitharaja.pdf (Accessed on 17 September 2021)
8. Bhattacharya, U.; Ghosh, S.K. & Parui, S. A two stage recognition scheme for handwritten Tamil characters. *In* 9th International Conference on Document Analysis and Recognition (ICDAR 2007), 23-26 September 2007, Curitiba, Brazil. pp. 511-515. IEEE. doi: 10.1109/ICDAR.2007.4378762
9. Tomar, A.; Choudhary, M. & Yerpude, A. Ancient Indian scripts image pre-processing and dimensionality reduction for feature extraction and classification: A survey. *Int. J. Comput. Trends Technol. (IJCTT)*, 2015, **21**(2), 101-24. doi: 10.14445/22312803/IJCTT-V21P1116
10. Smith R. An overview of the Tesseract OCR engine. *In* 9th International Conference on Document Analysis and Recognition (ICDAR 2007), 23-26 September 2007, Vol. 2, IEEE pp. 629-633. doi: 10.1109/ICDAR.2007.4376991
11. Stuner, B.; Chatelain, C. & Paquet, T. Handwriting recognition using cohort of LSTM and lexicon verification with extremely large lexicon. *Multimedia Tools Appl.,* 2020, **79**(45), 34407-27. doi: 10.48550/arXiv.1612.07528
12. Doermann, D. & Tombre, K. Handbook of document image processing and recognition. Springer Publishing Company, Incorporated; 21 May 2014. https://link.springer.com/referencework/10.1007/978-0-85729-859-1. (Accessed on 21 September 2021).

CONTRIBUTORS

Ms Monisha Munivel is a Research Scholar in the Department of Computer and Engineering, Sri SivaSubramaniya Nadar

College of Engineering, Chennai, India. She received her Master's degree in Network Engineering from Anna University, Coimbatore. Her research interests including: Digitisation of ancient scripts, cultural informatics and heritage, image processing, networking.
For this study she has investigated historical and technical data, methodology, software tool, and validation.

Dr V.S. Felix Enigo is working as an Associate Professor in the Department of Computer Science and Engineering, Sri SivaSubramaniya Nadar College of Engineering, Chennai, India. She obtained PhD in Computer Science from Anna University, Chennai. Her current field of interests include: Image processing and data science.
She has guided the research, reviewed the draft and finalised the paper.