Document Text Analysis and Recognition of Handwritten Telugu Scripts

1st Ashlin Deepa R. N
Department of Computer Science and Engineering
Gokaraju Rangaraju Institute of Engineering and Technology
Hyderabad, India
deepa.ashlin@gmail.com

2nd Y. Vijayalata
Principal
KG Reddy College of Engineering and Technology
Hyderabad, India
vijaya@ieee.org

3rd Atul Negi
Department of Computer Science and Engineering
University of Hyderabad
Hyderabad, India
atul.negi@uohyd.ac.in

2022 IEEE 4th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA) | 978-1-6654-6246-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCCMLA56841.2022.9989012

***Abstract*— Handwritten text recognition is an open problem of great interest in the area of automatic document image analysis. Handwriting recognition has been studied for a long time with only few practicable results when written on paper. The tasks related to text recognition becomes complex when it comes to Indic languages due to large size of character set. One such language is Telugu where the character list involves 150 unique characters. This paper addresses the procedure to recognize the Telugu handwritten document and convert the text into machine understandable language. The model is built on the ResNet-18, DBSCAN clustering algorithm for detecting the words in the hand written text image. The CNN, RNN and CTC. The proposed model is then evaluated using the IAM Telugu handwritten image dataset and few government school handwritten notes. The proposed model achieves an accuracy of 72.6% with a character error rate of 11.4%.**

Keywords—Document Text Analysis, Handwritten Text recognition, ResNet-18, DBSCAN, CNN, RNN, CTC

I. INTRODUCTION

Humans have been developing their communication skills in various forms of communication, such as speech and handwriting. The use of handwriting has been regarded as the most effective way of interacting with others. This has led to the development of new research areas in the field of text recognition. Due to the complexity of handwriting, it is very challenging to recognize handwritten documents in Indic languages, such as Telugu. Also, due to the varying strokes and nuances of handwriting [1], it is not feasible to perform a comprehensive analysis of these documents in offline or online formats.

Online text recognition is easier compared to offline text recognition as the styles, strokes, and font styles of each individual vary [2]. However, it is still very challenging to recognize text in offline mode due to the varying complexity of handwriting. When it comes to recognizing the text in Indic scripts, it is not easy to perform a comprehensive analysis of these documents. Optical Character Recognition (OCR) systems are available for languages like English, Latin, Roman, Devanagari, Bengali, Urdu languages Indic languages, such as Telugu and Tamil, have diacritic (Matras) symbols on the letters. These are used to change the sound of the word or letter. However, unlike in English, these symbols also make it hard to recognize text. The modern English alphabet has 26 letters, while Indian languages have more than 100 letters. Indian languages such as Telugu consists of 16 vowel symbols, 41 consonant symbols and 3 vowel modifiers. The consonants are also combined with the vowels which are orthographically represented using symbols, also known as “Matras”.

To promote the Indian languages and technology standardization, TDIL (Technology Development for Indian Languages) has been initiated by Ministry of Electronics and Information and Technology [MeitY][3]. The program also aims to create and access multilingual knowledge resources. E-Aksharayan [4] is a software that can be used to convert scanned or printed documents into an editable text. During the pandemic, the government created a solution called DIKSHA [5], which was designed to help the teachers share the materials with the students. This is an online platform that gives students access to a large number of materials created by teachers. Textbook materials provided by NCERT are available free of charge at DIKSHA and are accessible to students, teachers and parents. Rural schools can also use OCR to improve the communication and teaching skills of their students. Most of the schools in rural areas use the regional language as their medium of instruction. The students prefer to have notes made by the teachers as it allows them to improve their understanding of the subject. Handwritten notes are shared to the students in the form of images by scanning through mobile scanners or other scanners. The text present in these images are not clear to the student due to their variation in handwriting.
The objective of this work is to develop an OCR system for Telugu handwritten script which can convert Telugu handwritten notes into editable text (Fig.1), IAM Telugu dataset and teacher-created handwritten materials collected from Telugu medium schools are used to develop an OCR system that can be used to convert Telugu handwritten text into fully editable text.

One of the potential applications of document text recognition is postal automation, which allows users to collect information from various languages [6]. It can also be used in extracting data from bank applications or documents. In addition, it can be used in the digitization of historical

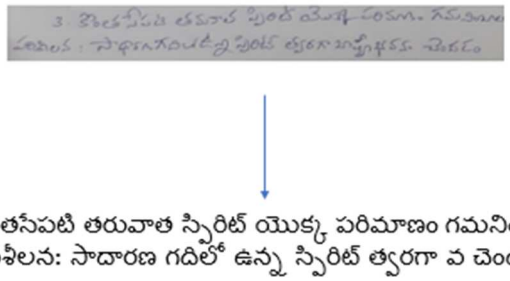

Fig.1. Conversion of handwritten text into editable text

documents, for instance by creating a digital library that stores information related to various medical conditions [7].

II. Related work

In recent years, several attempts have been undertaken to identify handwritten text in various scripts. To categorize handwritten English words, Sueiras et al. [8] suggested using segmentation-based high - level features with long short-term memory. The word image was jumbled by Dasgupta et al. [9] using the Arnold transform, and directional characteristics were then extracted from the jumbled image using the Hough transform. They used a multi-class support vector machine (SVM) to distinguish cursive words, and the CENPARMI English legal amount word collection, which contains 7744-word pictures of 32 distinct word classes, were used to assess their solution. Gorgevik et al. [10] created a model for Roman digit recognition that has 97% accuracy, using SVM. The model derives four characteristics from the image, including the ring-zone, contour profiles, histogram projections and kirsch characteristics.

A CNN model for Malayalam word recognition was proposed by Jino et al. in 2019 [11], and it was verified uing a dataset containing 314-word classes. Barua et al [12] recommended using the sequential minimal optimization (SMO) classifier with the HOG feature descriptor to recognize handwritten Bangla city names. Their method yields an accuracy of 90.65 percent across a database of 10,000 samples tested from 20 distinct city names. Bhowmik et al. [13] proposed a comprehensive approach employing a shape-based hybrid feature descriptor that integrates elliptical, tetragonal, and vertical pixel density histogram-based features. For recognition, MLP and SVM are both used independently.

L. Zhou et al [14] recognized Bangla and English as scripts adopting linked component profile-based characteristics among the works on Indic scripts. Line, word, and character-level work were accomplished. Using rotation-invariant texture characteristics based on multi-channel Gabor filtering and a gray level co-occurrence matrix. Singhal et al. [15] detected Roman, Devanagari, Bangla, and Telugu characters from line-level handwritten document pictures. Using characteristics such as horizontal and vertical centroids, sphericity, aspect ratio, white holes, etc., Hochberg et al. [16] identified six Indian and non-Indian scripts, including Arabic, Chinese, Cyrillic, Devanagari, Japanese, and Latin. Roy et al. [17] used component-based features, fractal dimension-based features, circularity-based features, and other techniques to identify six common Indian scripts, including Bangla, Devanagari, Malayalam, Urdu, Oriya, and Roman. Arabic and Latin characters were recognized from line-level handwritten documents by Moussa et al. [18] using fractal-based characteristics. Using a texture-based approach, M. Hangarge et al. [19] detected the Roman, Devanagari, and Urdu scripts. At the block level, the task was done. Considering Roman, Devanagari, and four south Indian scripts—Kannada, Telugu, and Tamil, Hangarge et al. [20] suggested a word level script recognition approach. Fuzzy sets were a notion that Sural & Das in 1999, utilized to recognize Bangla writing. They have constructed fuzzy sets on the character pattern pixels hough transform, from which other fuzzy sets may be created using t-norms, or intersections on the fundamental fuzzy sets [21]. Characters of the Bangla script may be identified using a multi-layer perceptron (MLP) classifier that had been trained using a variety of linguistic set memberships obtained from these t-norms.

The identification of isolated and continuous printed multi-font Bangla characters had been taken care of by Mahmud et al.,[22] in 2003. Segmentation at various levels, noise reduction, and scaling are all parts of pre-processing. Scaled symbols were used to construct Freeman chain code (Freeman 1974), which was then used to provide a set of feature vectors that were discriminating for the recognizer. Feed-forward neural networks are used for classification.

Majumdar in 2007, employed the digital curvelet transform and K-nearest neighbor classifier for identifying Bangla multi-font basic letters.[23] For feature extraction, the curvelet transform is employed. A group of K-nearest neighbor classifiers are trained using the curvelet coefficients of both the original and morphologically changed versions of a picture. To make a judgement, the output values of these classifiers are fused using a simple majority voting system.

Shanthi and Duraiswamy [24] discussed some of the preprocessing steps performed before recognizing Tamil handwritten characters. Chinnuswamy and Krishnamoorthy [25] used a linguistic approach to hand-printed Tamil characters. Hewavitharana and Fernando [26] used a two-step classification method to recognize Tamil characters. In the early stage, a certain character is classified into one of two groups: protagonist, descendant. The second step uses a statistical classifier for final identification. For the Telugu script, several works have considered machine-printed text recognition [27-28]. Some work on Telugu handwriting recognition online is also available in the literature [29].

The objective of this work is to help the teachers working in Government schools to convert the handwritten text notes to a fully editable computer friendly text which not only improves the quality of education, but also reduces the failure rates, which leads to the betterment of the students in rural areas. A better education also leads to improving the economic conditions of the rural areas.

III. Methodology

The model works in two phases Word Detection and Word Recognition. Phase-1, Word Detection, consists of detection of words using Algorithm 1. Architecture diagram for word detection is given in Fig.1.

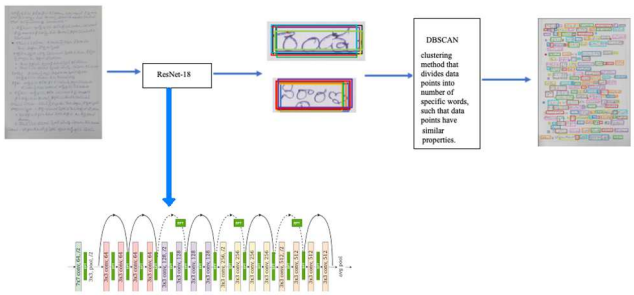

Fig.2. Architecture diagram for word detection phase.

This model follows U shaped ResNet18 architecture. When the image of size 448 x 448 is passed through layers of ResNet18, the feature maps then reduces the image to 14 x 14 and these layers upscale the maps step by step to size of 224x224 [30].

Algorithm 1:

Step-1: Load the image into the Model.

Step 2: The image is passed to ResNet-18 Neural Network Layers.

Step 3: Axis Aligned Bounding Boxes (AABB) are drawn around the word for every pixel of inner word.

Step 4: The AABBs are encoded by out-maps of model, 3 segmentation maps with encoding, 4 geometric maps.

Step 5: Apply clustering algorithm (DBSCAN). Co-ordinates of a single AABB are obtained.

The image is first passed to the ResNet-18 layer which identifies each pixel as a word (inner part or surrounding) or background pixel. For each pixel of inner word an axis aligned bounding box is predicted around the word and is encoded using segmentation maps and geometry maps [31], and so, multiple AABBs are predicted. The Jaccard distance between two AABBs is JD = 1 – IoU (intersection over union). A distance matrix including the Jaccard distances between all AABBs is calculated. Using this the clustering algorithm DBSCAN [32] enumerates AABB clusters. By considering median edge positions a single AABB is computed from cluster. The DBSCAN algorithm starts by selecting a point x from given dataset randomly and attach it to cluster1. Then it counts how many points are located within epsilon distance from x. If the quantity is greater than or equal to minPoints(n), then considers it as core point, then it will assign all these epsilon neighbors to same cluster. It will audit each member of cluster1 and find their following epsilon neighbors. The coordinates of a single optimum bounding box are obtained, which covers the entire content of the word.

The detected words are passed as inputs in phase-2, Word Recognition which follows Algorithm-2. Architecture diagram of word recognition is given in Fig.3.

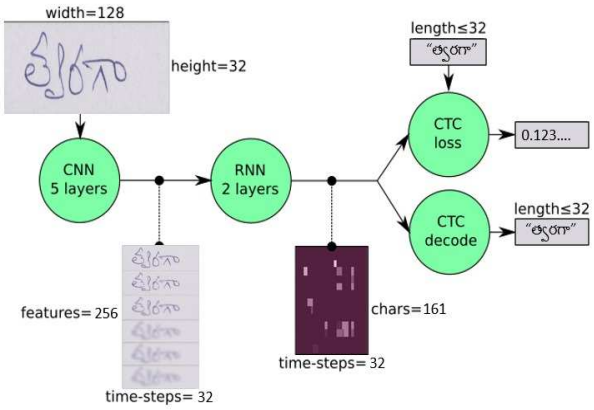

Fig.3. Architecture diagram for word recognition

Algorithm 2:

Step 1: The images of single words are loaded into the model.

Step 2: Image is fed to CNN layers to obtain a downsized matrix of size 32x256.

Step 3: The matrix is passed to RNN layers to obtain a character probability matrix of size 32x161.

Step 4: The character probability matrix is passed to the CTC layer to obtain the final decoded text. The loss value is calculated by comparing the output with ground truth.

Step 5: Loss value is calculated for images in the CTC layer.

The image obtained from phase-1 is fed as input to CNN layers. The purpose of CNN is to extract features from the image. The CNN consists of 5 layers and each layer consists of 3 operations. The Convolutional operation applies filter kernels.

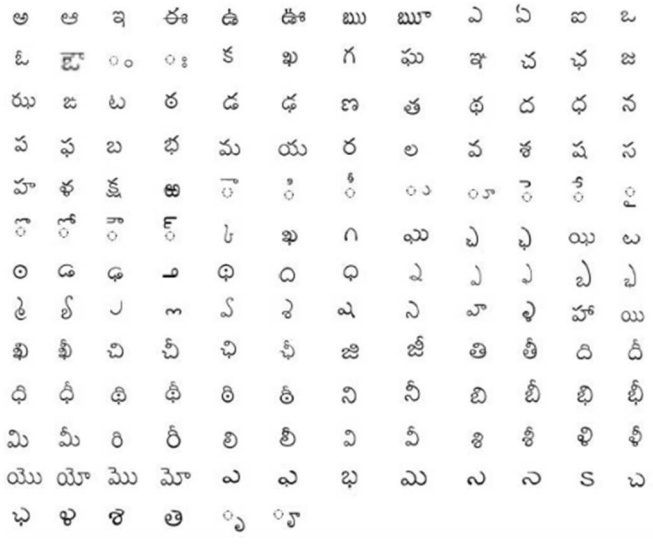

Fig. 4. Unique characters in Telugu Language

The Non-Linear ReLU function acts as a rectified linear operational unit and the final pooling layer downsizes the height of image and helps in reducing the computations. An output of size 32 x 256 along with 32-time steps is obtained. This output is now fed to RNN layer for sequential data propagation where the information propagates through the loop. The probability matrix of size 32 x 161 where 32 is maximum length of word and 161 characters (Fig.4: 150

unique characters of Telugu, 10 digits and CTC blank character) is obtained as output from RNN layers. The matrix is now passed to the CTC layer which performs two functions, encoding and loss calculation. The weights of the Neural Network are self-adjusted by CTC loss function. The combination of probability matrix and ground truth text is passed to CTC loss function. The issue of duplicate characters in encoded by pseudo character called blank. When encoding a text, random blanks are inserted at any position, that can be removed when decoding it. The loss value is calculated for images and text to train the Neural Network. Output contains a score for each character at each time step. The summation of character scores is 1 at each time-step. Loss is calculated by summing up all scores of possible combination of ground truth text. The aim is to train the NN such that we get output with a high probability.

Recognition of text in images can be achieved by approaches like Beam Search with character L-M, Beam Path Decoding and Word Beam Search. The first two approaches use only the output of NN and computes an approximation by taking the most likely character at each position. The most probable characters at each time stamp are connected which yields the best path. Then perform encoding by removing duplicate and blank characters. This gives us the final required text. This model uses Word Beam Search approach, which creates beams with scores. Initially we start with empty beam, then we add all possible characters to the first iteration and keeps the best score. This process is continued until the complete text is decoded. A beam may be in word-state or in non-word state. The algorithm shifts from word state to non-word state when a word is completed and also from non-word state to word state is also allowed. A prefix tree is used to show that which characters after adding enough form a word. Labels of edges are collected.

IV. Analysis

The IAM Telugu handwritten image dataset of IIIT-HYD, consists of 118,734 images. The dataset was split into 90% for training data, 10% for validation, and handwritten notes collected from the schools are taken as test cases. The total unique words in the dataset are 12,948. In the word detection model, total number of epochs is set to 10 with a batch size of 25. The main challenge with training neural networks is choosing the right number of training epoch. Training data with a large number of epochs can lead to overfitting, while training data with a small number of epochs can lead to underfitting. To overcome this obstacle, the early stopping approach is employed. Early stopping is a technique for stopping approach is employed. Early stopping is a technique for stopping training when the model's performance on a holdout validation dataset stops improving. In word-recognition model, total number of epochs is set to 15 with a batch size of 100, the model dumps the weights into screenshot.

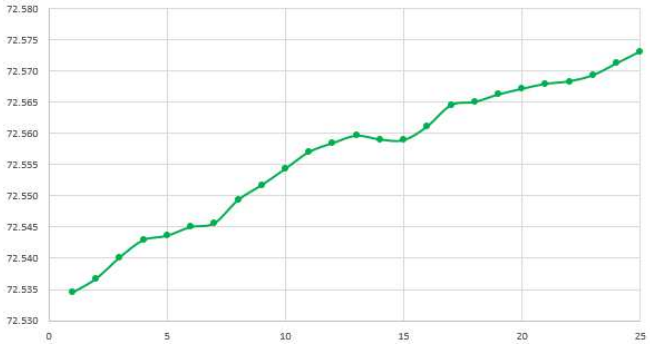

Fig.5. Graph representing Epoch vs Accuracy.

The number of epochs is represented on x-axis, while the accuracy is represented on y-axis. The model was run for 25 iterations. Also, Fig.5 shows that the model accuracy is gradually increasing. This work is limited to 25 epochs, which when tested, resulted the correct recognition by 72.6% accuracy.

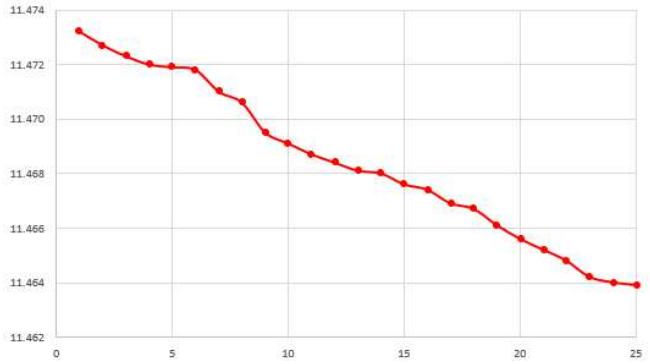

Fig.6. Graph representing Epoch vs CER.

Fig.6 demonstrates the character error rate calculated by comparing the decoded text to ground truth during the training process. The number of epochs, CER values are represented on x and y-axis respectively. It is observed that the CER values gradually decrease when trained against every epoch. This work is limited to 25 epochs which
The issue faced in detecting the words is often caused due to overlapping words. Fig.6a represents the overlapping of two different bounding boxes into single bounding box and Fig.6b represents the issue of single words being divided into multiple bounding boxes. This issue can be resolved by training and tuning the model with a greater number of manually collected handwritten notes.

Fig. 6a,6b Overlapping problem

V. Conclusion

A Telugu Handwritten text recognizing system is proposed in this work which recognizes the Telugu handwritten script and provides the content of the script in machine editable format. From the experimental results, we found that the

proposed model provides acceptable performance. The accuracy of the overall system reached 72.6%. This model could be improved by collecting more data for fine tuning the model.

ACKNOWLEDGMENT

We thank AudIntel India Private Limited, for helping us in collecting the handwritten Telugu data for the study.

REFERENCES

[1] Bhowmik, S., Malakar, S., Sarkar, R., *et al.*: 'Off-line Bangla handwritten word recognition: a holistic approach', *Neural Computing and Applications*, 2019, **31**, (10), pp. 5783–5798

[2] Tamen, Z., Drias, H., Boughaci, D.: 'An efficient multiple classifier system for Arabic handwritten words recognition', *Pattern Recognit. Lett.*, 2017, **93**, pp. 123–132

[3] https://tdil.meity.gov.in

[4] http://tdil-dc.in/

[5] https://diksha.gov.in

[6] Pal, U., Chaudhuri, B.: Indian script character recognition: a survey, *Pattern Recognit.*, 2004, **37**, (9), pp. 1887–1899

[7] Sampath, A., Gomathi, N.: Fuzzy-based multi-kernel spherical support vector machine for effective handwritten character recognition, *Sādhanā*, 2017, **42**, (9), pp. 1513–1525

[8] Sueiras, J., Ruiz, V., Sanchez, A., *et al.*: 'Offline continuous handwriting recognition using sequence to sequence neural networks', *Neurocomputing*, 2018, **289**, pp. 119–128

[9] Dasgupta, J., Bhattacharya, K., Chanda, B.: 'A holistic approach for off-line handwritten cursive word recognition using directional feature based on Arnold transform', *Pattern Recognit. Lett.*, 2016, **79, (C)**, pp. 73–79

[10] GorgevikD,CakmakovD,RadevskiV.Handwrittendigitrecognitionbycombiningsupportvectormachinesusingrule-basedreasoning. In: and others, editor. In Proceedings of the 23rd International Conference on Information Technology Interfaces. IEEE. 2001;p. 139–144. Available from: doi:10.1109/ITI.2001.938010.

[11] Singh, C., Bhatia, N., Kaur, A.: 'Hough transform based fast skew detection and accurate skew correction methods', *J. Pattern Recognit.*, 2008, **41**, (12), pp. 3528–3546

[12] Barua, S., Malakar, S., Bhowmik, S., *et al.*: 'Bangla handwritten city name recognition using gradient-based feature'. Proc. 5th Int. Conf. on Frontiers in Intelligent Computing: Theory and Applications, Singapore, 2017, pp. 343– 352

[13] Bhowmik, S., Malakar, S., Sarkar, R., *et al.*: 'Off-line Bangla handwritten word recognition: a holistic approach', *Neural Computing and Applications*, 2019, **31**, (10), pp. 5783–5798

[14] L. Zhou, Y. Lu , C. L. Tan, "Bangla/English Script Identification Based on Analysis of Connected Component Profiles", Lecture Notes in Computer Science, Volume 3872/2006, 24354, 2006, DOI: 10.1007/11669487_22.

[15] V. Singhal, N. Navin, D. Ghosh, "Script-based Classification of Hand-written Text Document in a Multilingual Environment", Research Issues in Data Engineering, pp.47, 2003.

[16] J. Hochberg, P. Kelly, T. Thomas, L. Kerns, "Automatic Script Identification from Document Images Using Cluster-based Templates", IEEE Trans. on Pattern Analysis & Machine Intelligence, vol. 19, no. 2, pp. 176- 181, 1997.

[17] K. Roy, S. K. Das, S. M. Obaidullah, "Script Identification from Handwritten Document", In Proceedings of The third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), Hubli, Karnataka, pp. 66-69, 2011.

[18] S. B. Moussa, A. Zahour, A. Benabdelhafid, A.M. Alimi, "Fractal-Based System for Arabic/Latin, Printed/Handwritten Script Identification", In Proceedings of International Conference on Pattern Recognition, pp. 1- 4, 2008.

[19] M. Hangarge, B. V. Dhandra, "Offline handwritten script identification in document images", International Journal of Computer Application, 4(6), pp. 6-10, 2010.

[20] M. Hangarge, K. C. Santosh, R. Pardeshi, "Directional Discrete Cosine Transform for Handwritten Script Identification", In Proceedings of 12th International Conference on Document Analysis and Recognition, pp. 344-348, 2013.

[21] Sural S and Das P K 1999 An MLP using Hough transform based fuzzy feature extraction for Bengali script recognition. *Pattern Recogn. Lett.* 20: pp.771–782

[22] Mahmud J U, Raihan M F and Rahman C M 2003 A complete OCR system for continuous Bengali characters, In: *Proceedings of the TENCON*, pp.1372–1376

[23] Majumdar A 2007 Bangla basic character recognition using digital curvelet transform. *Journal of Pattern Recognition Research* 2: 17–26

[24] Negi, A., Bhagvati, C., Krishna, B.: An OCR System for Telugu. In: Proc. ICDAR, pp. 1110–1114 (2001)

[25] Sukhaswami, M.B., Seetharamulu, P., Pujari, A.K.: Recognition of Telugu Characters Using Neural Networks. Int. J. Neural Syst. 6, 7–357,1995.

[26] Swethalakshmi, H., Jayaram, A., Chakraborty, V.S., Sekhar, C.C.: Online Handwritten Character Recognition of Devanagari and Telugu Characters using Support Vector Machines. In: Proc. 10th IWFHR, pp. 367–372 ,2006.

[27] Shanthi, N., Duraiswamy, K.: Preprocessing Algorithms for Recognition of Tamil Handwritten Characters. In: 3rd Int. CALIBER 2005.

[28] Chinnuswamy, P., Krishnamoorthy, S.G.: Recognition of Hand Printed Tamil Characters. Pattern Recognition 12, pp.141–152,1980.

[29] Hewavitharana, S., Fernand, H.C.: A Two Stage Classification Approach to Tamil Handwriting Recognition. In: Tamil Internet, California, USA , 2002

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Identity mappings in deep residual networks," in *ECCV*, 2016.

[31] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang ," east: an efficient and accurate scene text detector", Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, pp. 5551—5560, 2017.

[32] Gregory Axler, Lior Wolf , "Toward a dataset-agnostic word segmentation method", 25th IEEE International Conference on Image Processing (ICIP), pp. 2635-2639, 2018.