Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 171 (2020) 514–523

Procedia Computer Science

www.elsevier.com/locate/procedia

Third International Conference on Computing and Network Communications (CoCoNet'19)

Gujarati Handwritten Character Recognition from Text Images

Jyoti Pareek,Ph.D,
Department of Computer Science
Gujarat University,
Ahmedabad.
India.

Dimple Singhania,Msc. AI & ML,
Department of Computer Science,
Gujarat University,
Ahmedabad,
India

Rashmi Rekha Kumari, Msc. AI & ML,
Department of Computer Science,
Gujarat University,
Ahmedabad,
India

Suchit Purohit,Ph.D,
Department of Computer Science,
Gujarat University,
Ahmedabad
India

Abstract

Today is the era of paperless office and governance. It comes with numerous advantages like increased productivity and efficiency, pervasiveness, storage optimization, robustness and eco-friendliness. Hence there is a need of converting paper documents into machine editable form. This leads to development of OCR (Optical Character Recognition). OCR is a technique to convert, mechanically or electronically an image, photo or scanned document of a handwritten text (HCR-Handwritten Character Recognition) or printed text (PCR- Printed Character Recognition) into digital text. HCR is a form of OCR that is specifically designed to recognize the handwritten text whereas PCR focuses on recognition of printed text. HCR is more challenging as compared to PCR due to diversity in human writing styles, size, curve, strokes and thickness of characters. Based on data acquisition mode, the OCR can either be online or offline. Offline recognition is performed in two ways: handwritten and printed [1]. In offline mode, the characters are on paper and captured using scanner or high-resolution camera whereas in online mode the pixel values of characters are captured by movement of cursor, pen or stylus. The HCR systems are readily available for foreign languages and many of the Indian languages like Bangla, Devanagari and Gurumukhi but for Gujarati language the HCR development is still in its infancy stage. This study focuses on development of an artificial intelligence based offline HCR system for Gujarati language. Important contribution of this study is data collection, of size 10,000 images from 250 number of people, of different age groups, of different professions. This paper describes a supervised classifier approach based on CNN (Convolutional Neural Networks) and MLP (Multi-Layer Perceptron) for recognition of handwritten Gujarati characters. A success rate of 97.21% is obtained using CNN and 64.48% using MLP. Lot of work has been done at character level, but very few has been done at word level recognition.

Peer-review under responsibility of the scientific committee of the Third International Conference on Computing and Network Communications (CoCoNet'19).
10.1016/j.procs.2020.04.055

Major focus of this study was on creating a continuous workflow for image to text conversion at word level.

Peer-review under responsibility of the scientific committee of the Third International Conference on Computing and Network Communications (CoCoNet'19).

Keywords: Histogram projection profile; Connected Components Labelling; Segmentation; CNN; MLP; Optical Character Recognition.

1.Introduction

In a country like India where multiple languages are spoken and different Scripts are used, there is a need for the digitization of books and documents for every language. The purpose of our work is to make Offline Gujarati Character Recognition for converting scanned images of handwritten text (Fig.1) into digital text or machine editable text (Fig.2). The performance of this system is directly dependent on the quality of input documents. If the paper is crumbled or handwriting is not legible or the quality of image is not good, the accuracy of the system may decrease. OCR has a wide range of applications in government and business organizations, as well as individual companies and industries. There are many applications including automating Library and office documents, forms and bank check processing, Document reader systems, etc., to reduce the manual effort in converting handwritten text document to digital document and creating a Paperless office/ Governance. OCR of handwritten documents is still a research area and a challenge because of the unstructured and variable writing style of different people. Rest of this section describes Dataset generation method, and in upcoming sections the details of, methodology used, results and an idea on future work, has been discussed. References are given in last section.

1.1 Gujarati Script

Gujarati is the official language of Gujarat and is spoken by 60.3 million people[2]. It is derived from Devanagari script. In Gujarati script (Fig.3), there are 11 vowels and 36 consonants[3], few additional characters are also there. Consonant-vowel combination is denoted by attaching symbol of vowels to the consonants. Every vowel can be represented by a unique symbol, called vowel modifiers. Vowel modifier can appear to the right, left, top, bottom and middle of the character. A combination of two or more characters can make new characters (Conjuncts - Jodakshar). Unlike many North Indian languages, Gujarati has no shirorekha (Upper Horizontal line on the top of a word, as in Devanagari script) and hence all the characters in a word are isolated. The character set of Gujarati is almost double than that of English language. These are the few characteristics of Gujarati script which can be considered as a reason for the slow progress in development of Gujarati character recognition. [4]

Figure 1. Input Sample Image

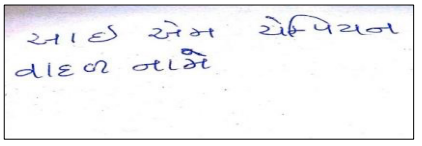


Figure 2. Output Image

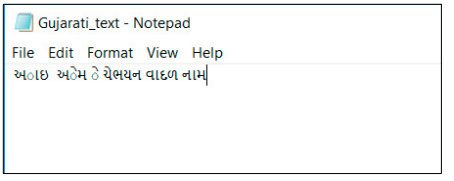


Figure 3. Gujarati Script[3]

Consonants
ક ખ ગ ઘ ઙ ચ છ જ ઝ ઞ ટ ઠ ડ ઢ ણ ત થ દ ધ ન પ ફ ભ બ મ ય ર લ વ શ ષ સ હ ળ ક્ષ જ્ઞ
Vowels
અ આ ઇ ઈ ઉ ઊ એ ઐ ઐ ઓ ઔ ઔ અં
Some Conjuncts
ક જ્ય ક્લ ચ્છ દ્ર ત્ર સ્પ સ્ત ર્વ ક્ક ત્ય ત્ર્ય ત્ત દ્ય
Consonant – vowel
જા ગી જી બુ ખૂ હે કુ કૈ પૈ પો ડૉ કૌ કં
Conjunct -vowel
જ્જા કે ક્લે ચ્છિ દ્વી ત્રુ સ્નૂ
Vowels modifiers
ા િ ી ુ ૂ ૃ ે ૈ ો ૌ ્ ઁ ં ઃ

1.2 Dataset Generation

To train a supervised classifier we need a labelled dataset consisting of images of all classes and its corresponding labels. Due to the lack of a benchmark dataset we have created our own dataset. We have collected data from people of different age groups and different professions. Till now we have a dataset collection of nearly 10,000 images. These images are then grouped into training and test set images of size 8000 and 2000 respectively, in an 80-20% ratio. The dataset is collected from different individuals and these images were scanned and then processed, to extract each character. The dataset images are pre-processed, binarized and segmented into individual characters of size 28 x 28 pixels and then each character is stored into its corresponding class folder. These dataset images are used to train our classifier. The binary image has characters (foreground) as 1 and background as 0. There is a total of 59 classes including vowels, modifiers (Matras), and consonants.

1.3 Related Work

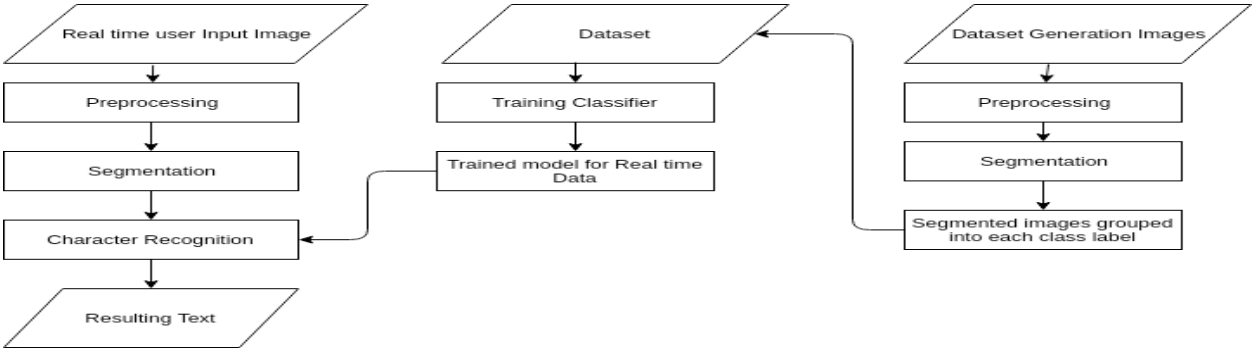
Patel, C., & Desai, A. in [5] describe segmentation of handwritten words into characters and modifiers, and methods for identification of zone boundaries for a word and usage of zone boundary details for segmenting words into its subcomponents. Connected component labelling is applied to detect subcomponents of a word, which can be further dissected if needed, to obtain other subcomponents of words. Magare, S. S., Gedam, Y. K., & Randhave, D. S. in [1] have presented different OCR techniques and its associated challenges in their paper. Sharma, A., Thakkar, P., Adhyaru, D. M., & Zaveri, T. H. [6] describes dataset collection , Zone based and projection profile based feature utilization and SVM and Naïve Bayes classifiers are used in their work. Recognition of isolated Gujarati handwritten characters is proposed using three different kinds of features and their fusion. Chain code-based, zone-based and projection profile-based features are utilized as individual features. Yang, Xuan., & Pu Jing in [7] focused on multi-digit handwritten numbers, trained a shallow CNN network and achieved an accuracy of 99.07% and utilized pre-processing and segmentation techniques to reduce the input image size fed into CNN. Jomy John, Pramod K. V., Kannan Balakrishnan [8] proposed scheme consists of two stages: a feature extraction stage, which is based on Haar wavelet transform and a classification stage that uses support vector machine classifier. A. A. Desai in [9] performs a Support vector machine comparison on different kernel techniques(Linear, radial, polynomial), with a Dataset size of 2000 samples of digits from 250 writers and achieved an accuracy of 92.60%, 95%, and 93.80% for linear, polynomial, RBF kernel respectively. Hetal R. Thaker,C. K. Kumbharana in [10], generated Structural features such as (No. of connected / disconnected components identified, no. of loops, No. of end points) in individual characters. Dataset of 5 characters (‘aNa’, ‘Ga’, ‘Sha’, ‘La’ and ‘Ja’), with

200 samples for each character and Decision tree was used. Overall accuracy of 88.78% is achieved. Success Ratio for 'aNa' is 98%, 'Sha' is 80.6%, 'Ga' is 94.66%, 'La' is 87.33% and 'Ja' is 83.33%.

2. Methodology

The methodology adapted for handwritten character recognizer is depicted in Figure 4. and techniques for each phase is summarized

Figure 4. The methodology followed in this system



2.1. Pre-processing

Pre-processing is the first and foremost step required for the Character recognition system. To convert the real time images in the same format as our dataset, and to reduce noise and remove unwanted background, the following pre-processing techniques are applied:

2.1.1 Scanning and Resizing

The image obtained from the user is scanned to convert it into digital image, the Region of Interest (ROI) is extracted by applying Canny edge detection followed by contour and bounding box detection, retaining only the largest contour with four corners (Fig.5). The ROI is transformed by aspect ratio and binary threshold is applied to convert into binary image. The binary image is then resized to have consistent image size. (Fig.6)

2.1.3 Noise Removal and Binarization

The image is subjected to Gaussian filter to remove noise and is converted to grayscale, using binarization by an adaptive threshold (A form of threshold that considers local spatial variations in illumination). (Fig.7)

2.1.4 Skew Correction

Handwritten text might be skewed to some extent (Fig.8) To perform skewness correction, skew detection is done by Run length smoothing and line image formation. Hough transform is then applied on the line image to obtain the skew angle of the text. For skew correction, the scanned image is rotated by that skew angle [2] (Fig. 9).

Figure 5. Image with unwanted background, showing largest contour with 4 corners.

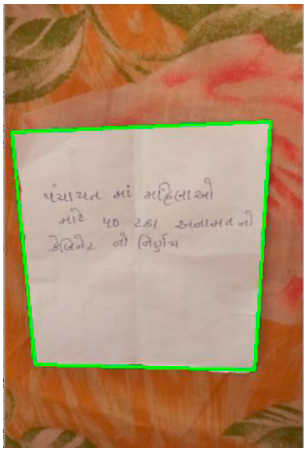


Figure 6. Output from scanning process

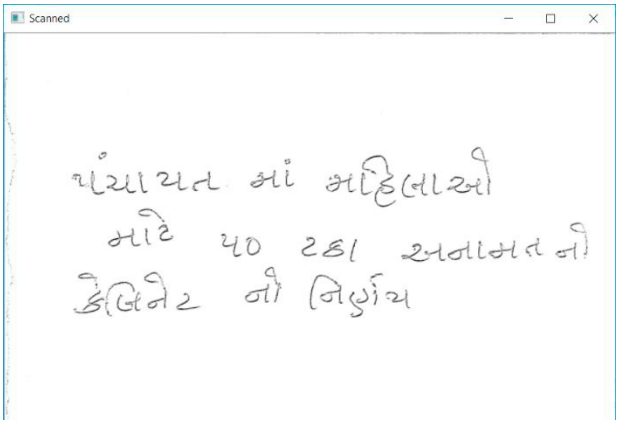


Figure 7. Results of Binarization

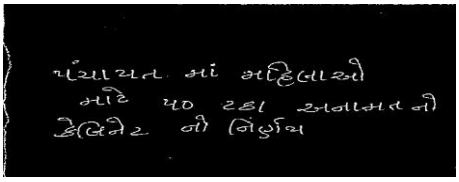


Figure 8. Image before skew correction

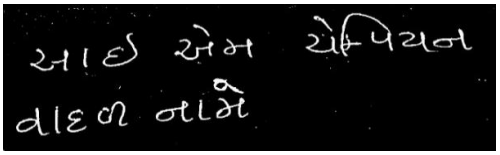
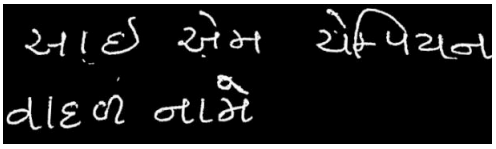


Figure 9. Image after skew correction



2.2. Segmentation

The pre-processed image needs to be segmented, to get individual characters from the entire text image [3]. Segmentation can be done by, identifying the sub-components, lines, and words and then character segmentation. These segmentation units are executed one after another. Segmentation techniques used in our work are described in following sub sections:

2.2.1 Connected Component Labelling

This method groups an image pixel into components based on pixel connectivity. The entire binary image is labelled. This algorithm scans the labelled image matrix and groups each pixel based on pixel connectivity, i.e., all pixel in a connected component shares similar pixel intensity values and are in some way connected with each other. Once all groups have been determined, each pixel is labelled with a grey level or colour (colour labelling) according to the component it was assigned to[5]. By using connected component labelling we determine the number of objects, the position of the objects, pixel area covered by the objects. The small

unwanted objects are removed by assigning 0 value to pixels for object labels whose area is less than 12% of the average size of all the objects (Fig. 10)[5].

2.2.2 Histogram Projection Profile

Another method for line segmentation is the histogram projection profile. This method calculates the number of ON pixels present for each row and then computes a histogram based on those values (Fig.13, Fig.15). The histogram is calculated based on the sum of ON pixels (pixels with value 1). A threshold is set, and a new image, of the size equal to the segments is generated from the image, based on the coordinates of the rows. The threshold decides which rows to segment from, if the previous row index has histogram value less than threshold and the next row index has histogram value greater than threshold, then the current row index is considered as upper boundary and if the current row index value has a value greater than the threshold and next row index has histogram value less than the threshold, then the row index is considered as lower boundary. These upper and lower boundary row coordinates are considered for line segmentation. This gives line segmentation images. For word segmentation, column wise histogram values are checked with the threshold and left, and right boundary is found. This forms the segmentation boundary for a word from the line image. (as shown in Fig.14, Fig.16)

2.2.3 Text Blob detection

To identify the regions where the text is present on the paper, we are using morphological operations such as dilation for text region detection. (Fig. 11) Different kernels are used to segment line region, word region, and character region. The contours over this dilated image gives X, Y coordinates to segment words from original line image.

2.2.4 Contour segmentation

The region where the text is present, is identified and then contour and bounding box are detected (Fig. 12). This bounding box is then separated, based on lines, words, and characters. The contours are not always in the order in which the text is written, so they are sorted based on X and Y coordinates of the identified contours, to get the contour images in the order in which the text appears on the image.

Figure 10. Results of connected component labelling

Figure 11. Text blob detection using Dilation

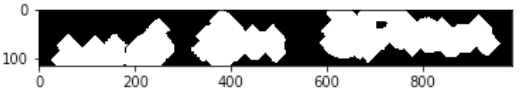


Figure 12. Contour and Bounding Box on the Word Image

Figure 13. Graph for horizontal histogram projection profile

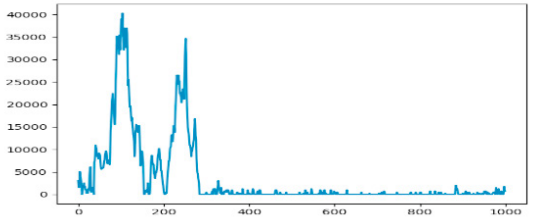


Figure 14. Result of Horizontal projection profile on Image

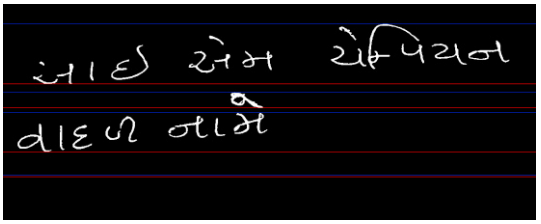


Figure 15. Graph for vertical projection profile

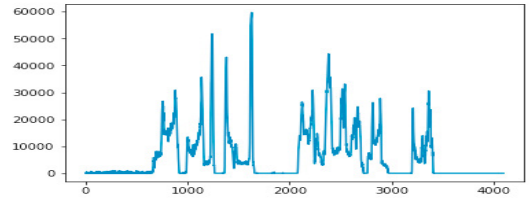
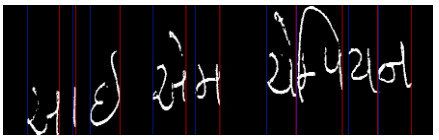


Figure 16. Result of vertical projection profile



2.3. Classification

The segmented content is fed to the classifier employing deep learning architecture using MLP and CNN models with dataset near to 10,000 images of 59 classes. The parameter tuning details of deep learning architecture is summarized in Table1. Due to small size of dataset, data augmentation such as rotation, shift and normalization has been done to the character dataset to increase size of the data during training. An Accuracy of 64.87% was achieved in MLP & 97.21% accuracy was achieved using CNN.

Table 1. Parameters of MLP and CNN

MLP Parameters			CNN parameters		
Layer(type)	Output Shape	No. of Parameters	Layer (type)	Output Shape	No. of Parameters
dense_5 (Dense)	(None, 512)	401920	(Input Layer)	(None, 28, 28, 1)	0
(Activation)	(None, 512)	0	(Conv2D)	(None, 28, 28, 6)	156
(Dropout)	(None, 512)	0	(Activation)	(None, 28, 28, 6)	0
(Dense)	(None, 256)	131328	(MaxPooling2)	(None, 14, 14, 6)	0
(Activation)	(None, 256)	0	(Conv2D)	(None, 14, 14, 16)	2416

(Dropout)	(None, 256)	0
(Dense)	(None, 59)	15163
(Activation)	(None, 59)	0

Total params: 548,411
Trainable params: 548,411
Non-trainable params: 0

(Activation)	(None, 14, 14, 16)	0
(Max Pooling)	(None, 7, 7, 16)	0
(Flatten)	(None, 784)	0
(Dense)	(None, 200)	157000
(Batch Normalisation)	(None, 200)	800
(Activation)	(None, 200)	0
(Dense)	(None, 59)	11859

Total params: 172,231
Trainable params: 171,831
Non-trainable params: 400

3.Results

Classification prediction was made on test data, in order to estimate the skill of the model, on unseen data using MLP and CNN models. The performance for both the models is summarized in Table2. The results of CNN and MLP are compared using graph (Fig. 17,18,19).

Table 2. Comparison of MLP and CNN results

MLP	CNN
Accuracy: 64.87% on test set and 92.10% on training set Loss: 2.04% Activation function: Sigmoid + Softmax (Last layer) Loss function: Binary cross entropy Optimizer: Adam	Accuracy: 97.21% on test data and 99.31% of accuracy on training data Loss: 0.04 % Activation function: RELU + Softmax (Last layer) Loss Function: Binary cross entropy Optimizer: Adam

Figure 17. Graph Showing Accuracy for MLP

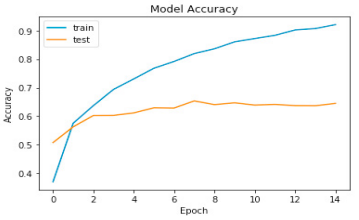


Figure 18. Graph Showing loss for MLP

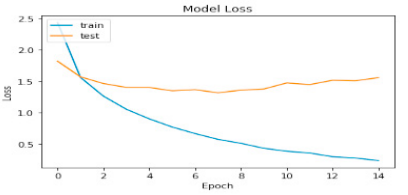
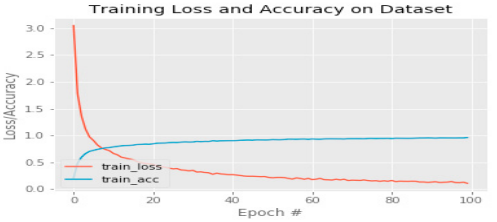


Figure 19. Graph showing Accuracy and loss for CNN



4.Post-Processing

The prediction contains many candidate characters, this can be corrected using two major approaches:

4.1 Grouping

The characters and its modifiers are grouped together in this step based on their location in the document

Algorithm:

1)Get the predicted Unicode character sequence as a list.

2)Check if the character Unicode falls in the range of Unicode defined for each modifier (Matra).

If the Unicode is for a modifier (Matra) that appears to the left of character, combine it with the next root character

If the Unicode is for a modifier (Matra) that appears to the right of the character, combine it with the previous root character

3)If the character has been appended with the modifier (Matra), remove it from the list so that we do not have compound character along with root character repeating itself.

4.2 Error identification and correction

After grouping of Characters, word level errors must be identified. One of the approaches is the use of dictionaries, which has proven to be the most efficient method for error identification and correction. Given a word, in which an error may be present, the word is looked up in the dictionary. If the word is not present in the dictionary, an error is identified and may be corrected by changing the word with the most similar word[2]. This method is not currently implemented in our work but is said to be one of the valid methods for spell checking in post-processing technique.

4.3 Text file generation and Unicode encoding

The prediction obtained is class labels, which are numbers. These numbers are mapped to its corresponding Gujarati Unicode

character. A Unicode mapping dictionary does this task. The sequence of characters obtained from grouping is then mapped to the corresponding value in the Unicode dictionary. The prediction is mapped to its corresponding Unicode and the Gujarati character is written into a text file. Now we have the image contents into the text file.

4.4 GUI

A simple GUI is built to bind all these processes and show only the desired output to the user (Fig. 20). The output is saved in the text file. (Fig. 21).

Figure 20. GUI for character recognition from images

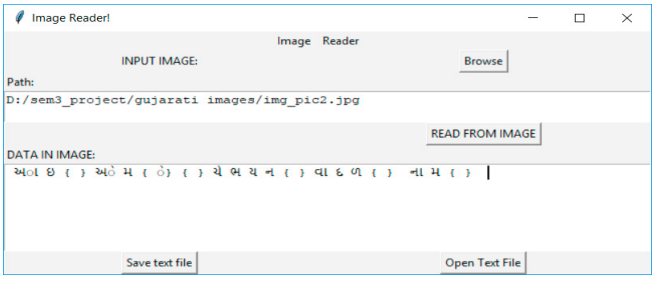
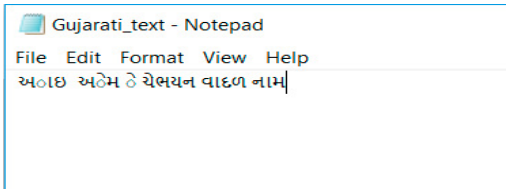


Figure 21. Text file with recognized characters



5. Conclusion

This paper shows an artificial intelligence based offline HCR system for Gujarati language. We found our system to be robust and accurate. The future work directs to enhance/extend the dataset to increase robustness against variability in different handwriting. Also handling of modifiers and special characters as individual classes and properly assigning them, their root characters and use of Natural Language Processing techniques to improve word level accuracy by correcting the possible wrong words in the recognized text.

6.References

[1] S. S. Magare, Y. K. Gedam, and D. S. Randhave, “Character Recognition of Gujarati and Devanagari Script : A Review,” *Int. J. Eng. Res. Technol.*, vol. 3, no. 1, pp. 3279–3282, 2014.

[2] A. G. Ramakrishnan, “Design and Development of Gujarati OCR Design and Development of Gujarati OCR Requirement for the Degree of Master of Engineering in,” no. June 2012, 2018.

[3] V. Patel and A. Pandya, “A Survey on Gujarati Handwritten OCR using Morphological Analysis,” vol. 2, no. 2, pp. 2395–1990, 2016.

[4] A. A. Jain and H. A. Arolkar, “A Survey of Gujarati Handwritten Character,” vol. 6, no. Ix, pp. 461–465, 2018.

[5] C. Patel and A. Desai, “Extraction of Characters and Modifiers from Handwritten Gujarati Words,” *Int. J. Comput. Appl.*, vol. 73, no. 3, pp. 7–12, 2013.

[6] “Features fusion based approach for handwritten Gujarati character recognition,” *Nirma Univ. J. Eng. Technol.*, vol. 5, no. 2, pp. 13–19, 2017.

[7] X. Yang and J. Pu, “MDig : Multi-digit Recognition using Convolutional Neural Network on Mobile,” pp. 1–10.

[8] J. John, K. V. Pramod, and K. Balakrishnan, “Unconstrained handwritten Malayalam character recognition using wavelet transform and support vector machine classifier,” *Procedia Eng.*, vol. 30, no. December, pp. 598–605, 2012.

[9] A. A. Desai, “Gujarati handwritten numeral optical character recognition through neural network,” *Pattern Recognit.*, vol. 43, no. 7, pp. 2582–2589, 2010.

[10] H. R.Thaker and C. K. Kumbharana, “Structural Feature Extraction to recognize some of the Offline isolated Handwritten Gujarati Characters using Decision Tree Classifier,” *Int. J. Comput. Appl.*, vol. 99, no. 15, pp. 46–50, 2014.

[11] Word level segmentation was performed from the following reference:
Subhrajyoti Sen,Shreya V prabhu,Steve Jerold,Pradeep JS
https://github.com/SubhrajyotiSen/KannadaHandwritingRecognition (2018)

7 . Acknowledgement

We are thankful to Department of Computer Science, Gujarat University for their support to carry out this research.