International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 13 Issue 6, June, 2024, Impact Factor: 8.285

An Automated Framework to Generate Handwritten Dataset for Gujarati Handwritten Character Recognition

Snehal Shukla[1], Dr. Purna Tanna[2]

Ahmedabad, India

ABSTRACT

Handwritten character Recognition is a popular area of research. Research on handwritten characters starts with collection of dataset. Handwritten characters can be collected from different individuals and the process of data fetching starts. Data can be fetched by manual efforts like manual scanning, segmentation and labeling for segmented characters. After scanning pre-processing steps need to be applied on the collected data. The proposed algorithm gives the cycle to robotize the arrangement of information assortment. It reduces efforts of manual segmentation and sorting of segmented data. No manual mediation is expected in the proposed framework.

Keywords: Handwritten Character Recognition, automatic segmentation, Binarization, Segmentation, zone detection

INTRODUCTION

Handwritten character recognition has become a critical area of research and development, propelled by the increasing digitization of information and the demand for efficient data processing. While considerable progress has been made in the recognition of printed text, the recognition of handwritten characters poses unique challenges, particularly when it comes to preserving and promoting cultural diversity. One such cultural and linguistic context that demands specific attention is the Gujarati script, an integral part of India's linguistic and cultural heritage.

Gujarati, spoken by millions in the Indian state of Gujarat and by diaspora communities worldwide, employs a distinct script that reflects the rich cultural and historical tapestry of the region. As the world witnesses an accelerating shift towards digital communication and information storage, the need for accurate and efficient Gujarati Handwritten Character Recognition (GHCR) systems becomes paramount for several compelling reasons.

As shown in Fig 1 Gujarati language is having 34 vyanjans, 13 swar and 15 different types. In Gujarati handwritten OCR, all these characters and different combinations of them should be identified. The implementation of a Gujarati Handwritten Character Recognition (GHCR) system benefits from a large and diverse dataset. A large dataset allows the system to encounter a diverse range of handwriting styles, including variations in size, slant, thickness, and curvature of characters.

This helps the GHCR system generalize well to recognize characters written by different individuals. Training a GHCR system on a large dataset aids in creating a model that generalizes well to unseen data. This is crucial for the successful recognition of characters in real-world scenarios where the system encounters diverse handwriting styles it hasn't seen during training.

Real-world scenarios often involve noise, distortions, or imperfections in the handwritten text. A large dataset that includes variations in writing conditions, paper quality, and writing instruments helps the system learn to handle such noise and still make accurate predictions.

Exposure to a substantial dataset promotes the development of a robust model. The system becomes more resilient to variations in writing style, allowing it to perform well on different kinds of handwritten samples.

A large dataset helps prevent overfitting, where the model becomes too specific to the training data and performs poorly on new, unseen data. This is particularly important for developing a GHCR system that can accurately recognize diverse handwriting styles.

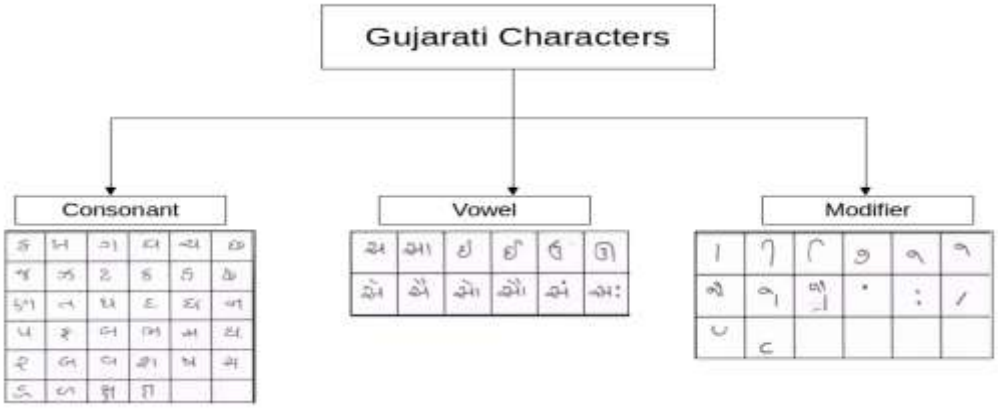

Fig 1: Gujarati alphabets with vyanjan, swar and matra

LITERATURE REVIEW

A huge number of researchers have worked on handwritten character recognition. The dataset generation is primary need of the process. The Process of generating dataset plays major roll in recognition process.

Urdu language is also one of the widely used language, Sajid Anwar et al [7] have proposed a framework for handwritten urdu characters and numbers. Classification using neural network provides accurate results. The framework uses CNN and DNN for classification of data. According to the authors, Segmentation and labeling is tedious task in the whole process but it is very much helpful.

Ashok Kumar Pant et. al. [8] works for handwritten Devanagari character set that can be used to write Hindi, marathi, sanskrit, nepali etc. They have used manual process of generation of dataset. Collected data from different indiviuals and cropped the image as per requirements. Dataset of 92,000 images was produced.

El-SawyA[9] have published dataset of 16,800 characters using CNN classification technique. Dataset is scanned and segmented automatically using matlab2016a. CNN gives them better result as compare to other techniques.

LeCun Y [10] used Modified NIST set as their dataset and used letNet-5 for classification of english numeric characters.

Liu, C. L. et. al. [11] uses Chinese Language dataset CASIA and classified by Modified Quadratic Discriminant Function and Nearwst Prototype Classifier with Discriminative Feature Extraction.

Naz S, Hayat et. al. [12] have used dataset provided by Center for Pattern Recognition and Machine Intelligance, Canada for Urdu like Cursive offline character recognition.

Ahmed et. al. [13] also have worked with Urdu handwritten Character set and used one dimensional BLSTM classification algorithm.

Sagheeret. al. [14] have worked with offline urdu character dataset and produced a dataset for further research.

Rabby, A. S. A. et. al. [15] have described the way to generate dataset for any language characters. The Form is used to collect data, scanned the form and then apply cropping using canny edge detection algorithm to find proper form, then row and column wise cropping is done. OTSU algorithm is used to smoothing and noise removing process.

Thaker, H. R. et. al. [16] have worked with Gujarati handwritten character for dataset generation. The proposed algorithm gives 94% accuracy in fetching data from datasheet.

K. K. M. BAHETI et. al. [17] have mentioned that we do not have any specific dataset for gujarati handwritten characters. They have worked with data collected from 80 different users and applied K-Nearest Neighbor Algorithm for classification.

B. V. Dhandraet. Al [18] have worked with kannada, telugu and devanagri handwritten numerals and used novel approach for noise removal.

AnupamaSahu and Sarojananda Mishra [21] have worked on Odia Printed as well as handwritten characters recognition. According to them segmentation is very much important process for OCR and listed many problems faced during segmentation like character size, overlapping of characters, etc. They have used structural, topological and water reservoir principle to segment touching characters into individual characters. In future they are going to propose algorithm for autometed segmentation of overlapping character also.

Kamal Maro et al [22] have used neural networks for recognition of gujarati handwritten digits. They have used two approaches, feature extraction and skeletonization and then feature extraction. They are achieving 85.33% accuracy without skeleton, 80.5% accuracy with skeletonization. Classification work better with neural network according to them.

GururajMukarambiet. Al[23] have worked with handwritten kannada and english scripts and able to achieve 73.33% and 96.13% accuracy for recognition of characters. They have used SVM classifier with 2 folds to recognize Kannada characters. They have implemented single algorithm and they have thinning and slant of character rectification process in it.

By reseaching from literature, we can conclude that work for modifiers are not much explored. It can be explored and get result to read gujarati handwritten characters.

DATA PROCESSING

Data Gathering:
Data gathering is the most important process for HCR. We need data for training as well as testing. For training we need single characters and for testing whole paragraph is required. As many characters can be gathered, our HCR works better, because large set of trainning data can be given to the system. As shown in Fig 2 we have a blank form for data gathering. The form consist of 17 columns and 30 rows of boxes. It will give 510 boxes in one form. Each box can be filled with single character with any matra. The filled form is also visible in Fig 2. Data collection can be done from different set of people of different age group and gender. In this paper we are focusing on training dataset. Data can be collected using a form that can be divided in same sized boxes as shown in Fig2. This form is useful for the input for training dataset.

Dataset for training should be having different types of handwriting. For that we have distributed the form with people of different age groups. This will help to generate good dataset as all inputers will have different style of writing. Here, data is gathered in 100 forms from different users. The filled form must be segmented to get individual characters from the filled form to process further. Before segmenting the individual boxes to get single characters, image must be prepared to apply segmentation. As data is collected from different writers, it may contains different writting strokes, different types of noise. So, first step is to clean the image by applying binarization and thinning on it.

Image Binarization:
After getting filled form, to start process, scanned copy of form should be there. As the writers are different, they may use different color pens as well as the image with multiple colors is difficult to analyze. If the image is in one or two color, it will be easy to implement further operations on it. To convert image into gray scale, binarization can be used. Binarization is the process of converting image into binary form that means only black and white colors.

Jyotsana et al[20] defines binarization process as a pre-processing step for analysis and processing of image. They have compared different techniques of binarization on basis of PSNR, F-Measure, NRM, and MPM parameters, and came to conculsion that no single techniques is best to get best optput of binarization.

K.Ntirogiannis et al[1] defines binarization as the process that segments the document image into the text and background by removing any existing degradations. We have many types of binarization techniques like Gradient Based Thresholding, Niblack Method, Otsu Method, Nick Method, Bradley Method, Bersen Method, Local Adaptive Thresholding.

Wan Azani Mustafa et al [15] have compared all the methods but using any method is not accurate for binarization. They have found Fuzzy C-Means algorithm with 97.02% accuracy for binarization[2].

Fu Chang [3] proposed a new method named Hadamard multiresolution analysis to get improvement in the output of binarization for OCR.

Binarization can be implemented using thresholding. Thresholding is a process of converting the pixel into two value We have also used openCV to convert input image to binarization.

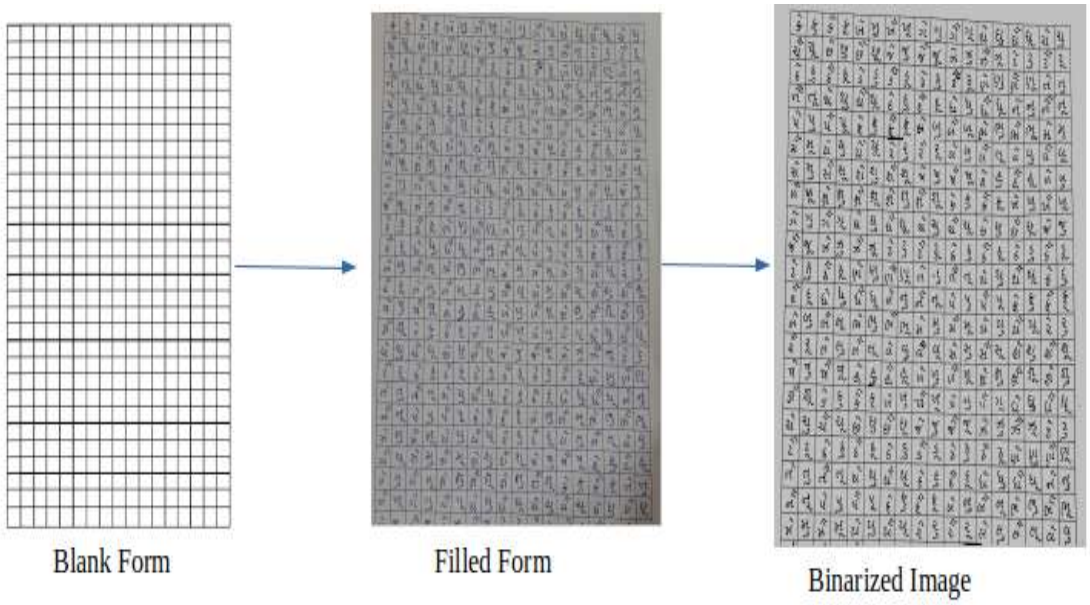

Fig2: Image of Blank form, Filled Form and Binarized form

Image Thinning and Noise Removal:
Next process is to apply thinning of image. As people who have given input, have used different point of pens, so the characters will be of different thickness.

For character recognition, characters should be of same thickness. To achieve same thickness, thinning of image must be implemented. It is a process of converting image into single pixel notation. It is useful in image compression, biomedical image analysis, printed circuit board analysis, fingerprint analysis, etc.[19] and it is considered as an essential step for many OCR[5].

Shaikh &shaikh[6] have listed the imitations of Lei-Hang's algorithm and proposed a modified algorithm for thinning process, which overcome the limitations of Lei-Hang's algorithm. We have implemented thinning using openCV library. Now our output image is ready for segmentation.

Image Segmentation:
First level of segmentation should be implemented to separate individual box and then each box should be segmented more to get characters and modifiers from each box. Segmentation can be implemented by various algorithms of neural network and machine learning. In this paper we have focus on segmentation of boxes. The proposed framework will be useful for generation of dataset for Gujarati Handwritten Character Recognition. Segmentation can be implemented in three manners: 1. Manual Segmentation, 2. Semi automatic segmentation and 3. Automatic Segmentation.

1. In manual segmentation, no need to implement any algorithm, we have to manually crop the image using any cropping tool and can apply directly recognition algorithms on segmented images. This process is not recommended because we have to work very precisely to get perfect segmented image. If the segmentation is not done properly, recognition can not be done properly.
2. In Semi automatic segmentation process, we need to submit properly cropped image to segmentation algorithm. Here we have to take care that image should not have any white space out side the line of box. If we fail to crop image perfectly, the other level of segmentation will be failed.
3. Proposed system is automatic segmentation system where we just have to give scanned image of the form. All other processes will be done automatically. We have collected data from different age groups to get different style of writing. People have used different pens so we got various pen strokes of various size. Color of pen ink are also not same. We have collected data from 100 people to get a large dataset. In one form, we have 17*30=510 boxes of single character. In total we have collected 510*100=51000 characters. Now the process is to apply segmentation to get individual characters from form and segment each character zone wise.

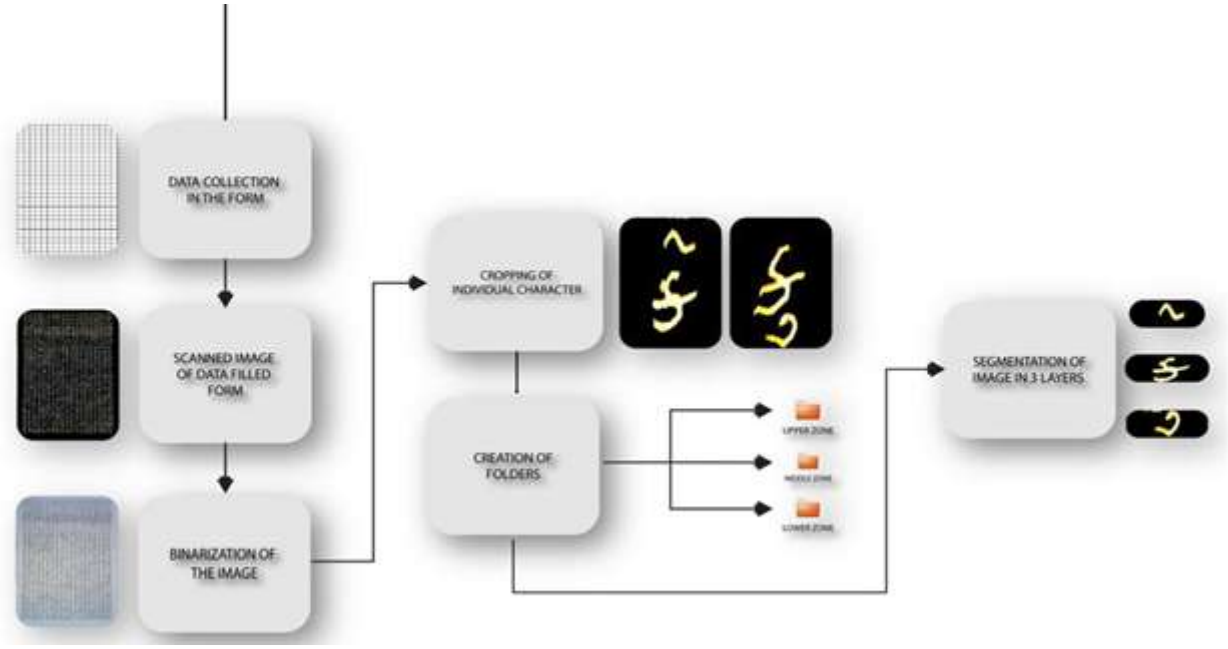

Fig 3: Framework for automatic segmentation of form

The form is designed to collect data. Now it is time to retrieve individual character from the form. Fig 3 Shows the framework for automatic segmentation of form. First we have to convert image into black and white image. Then we have to search outlines of boxes to segment box from the form. We can fetch individual box using following algorithm. Fig 4 Shows the segmentation of image.

Algorithm to fetch individual box from input form:
Step 1: Define Height = height of the page.
Step 2: Define Width = width of the page.
Step 3: finding the contour.
Step 4: Crop the image according to detected contour which gives square image of each characters..
Step 5: Save that cropped image with appropriate name and at proper location.
Step 6: Repeat the process till the last pixel.

The above algorithm is for cropping individual box from the whole page of input form. We will get images in a separate folder called Segmented Box with title b1. Jpg, b2.jpg and so on till last box. Now we need to segment the box further to fetch the samples for zone wise character parts. Gujarati Characters can be segmented in three zones that is upper zone, middle zone and lower zone. Fig 5, 6, 7 displays the final segmented image of character.

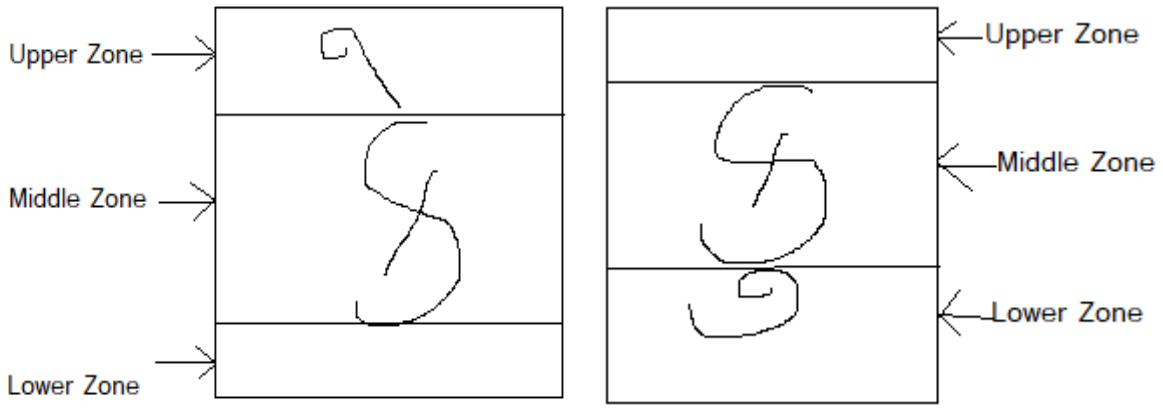

Fig 4: Segmentation of the Characters

Algorithm for Zone wise segmentation:
Step 1: Take single file from folder SegmentedBox.
Step 2: Convert the image into matrix of 0 and 1.

Step 3: Read each line of matrix and divide it into 3 zones.
Step 4: Crop the image in three zones and name it as follows:

I. Upper Zone image will be named as UZ<no> and added into folder named UpperZone.
II. Middle Zone image will be named as MZ<no> and added into folder named MiddleZone.
III. Lower Zone image will be named as LZ<no> and added into floder named LowerZone.

Fig 5: LZ1

Fig 6: MZ1

Fig 7: UZ2

Now we can apply recognition algorithms on segmented images. This process is completely automatic. We can get directly segmented images from one image of input form.

CONCLUSION

Dataset generation using proposed framework is fully automatic process. Manual approach takes extra time and efforts. Proposed framework provides segmentation from whole sheet to individual zone wise images in separate folders. This output can be further used for categorization of modifiers. The pure intention of generation of dataset is to implement HCR for Gujarati Handwritten characters, specially modifiers written in upper zone and lower zone. Here focus is on only on the modifiers of Upper and Lower zone therefore the data filled, contains only characters with some modifiers not all. We can broaden this process by taking characters with all types of modifiers.

REFERENCES

[1]. Ntirogiannis, K., Gatos, B., &Pratikakis, I. (2014). A combined approach for the binarization of handwritten document images. Pattern recognition letters, 35, 3-15.
[2]. Mustafa, W. A., Aziz, H., Khairunizam, W., Ibrahim, Z., Shahriman, A. B., &Razlan, Z. M. (2018, August). Review of different binarization approaches on degraded document images. In 2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA), 1-8. IEEE.
[3]. Chang, F., Liang, K. H., Tan, T. M., & Hwan, W. L. (1999, September). Binarization of document images using Hadamard multiresolution analysis. In Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318), 157-160. IEEE.
[4]. Steinherz, T., Intrator, N., &Rivlin, E. (2000). A special skeletonization algorithm for cursive words. In Proc. Int'l Workshop Frontiers in Handwriting Recognition, 529-534.
[5]. Lam, L., Lee, S. W., &Suen, C. Y. (1992). Thinning methodologies-a comprehensive survey. IEEE Transactions on Pattern Analysis & Machine Intelligence, 14(09), 869-885.
[6]. Shaikh, N. A., & Shaikh, Z. A. (2005, December). A generalized thinning algorithm for cursive and non-cursive language scripts. In 2005 Pakistan Section Multitopic Conference (pp. 1-4). IEEE.
[7]. Anwar, S., Mehrban, B., Ali, M., Hussain, F., & Halim, Z. (2021). A novel framework for generating handwritten datasets. Multimedia Tools and Applications, 80, 9657-9669.
[8]. Acharya, S., Pant, A. K., &Gyawali, P. K. (2015, December). Deep learning based large scale handwritten Devanagari character recognition. In 2015 9th International conference on software, knowledge, information management and applications (SKIMA), IEEE, 1-6.
[9]. El-Sawy, A., Loey, M., & El-Bakry, H. (2017). Arabic handwritten characters recognition using convolutional neural network. WSEAS Transactions on Computer Research, 5(1), 11-19.
[10]. LeCun, Y., Bottou, L., Bengio, Y., &Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
[11]. Liu, C. L., Yin, F., Wang, D. H., & Wang, Q. F. (2013). Online and offline handwritten Chinese character recognition: benchmarking on new databases. Pattern Recognition, 46(1), 155-162.
[12]. Naz S, Hayat K, Razzak MI, Anwar MW, Madani SA, Khan SU (2014) The optical character recognition of urdu-like cursive scripts. Pattern Recogn, 47(3), 1229–1248
[13]. Ahmed, S. B., Naz, S., Swati, S., &Razzak, M. I. (2019). Handwritten Urdu character recognition using one-dimensional BLSTM classifier. Neural Computing and Applications, 31, 1143-1151.

[14]. Sagheer, M. W., He, C. L., Nobile, N., &Suen, C. Y. (2009). A new large Urdu database for off-line handwriting recognition. In Image Analysis and Processing–ICIAP 2009: 15th International Conference Vietrisul Mare, Italy, September 8-11, 2009 Proceedings 15 , Springer Berlin Heidelberg, 538-546.
[15]. Rabby, A. S. A., Haque, S., Shahinoor, S. A., Abujar, S., & Hossain, S. A. (2018). A universal way to collect and process handwritten data for any language. Procedia computer science, 143, 502-509.
[16]. Thaker, H. R., &Kumbharana, C. K. (2014). Preprocessing and Segregating Offline Gujarati Handwritten Datasheet for Character Recognition. International Journal of Computer Applications, 97(18).
[17]. K. K. M. BAHETI M. J., "Comparison Of Classifiers For Gujarati Numeral Recognition," International Journal of Machine Intelligence, vol. 3, no. 3, pp. 160-163, 2011.
[18]. Dhandra, B. V., Benne, R. G., &Hangarge, M. (2010). Kannada, Telugu and Devanagari handwritten numeral recognition with probabilistic neural network: a novel approach. International Journal of Computer Applications, 26(9), 83-88.
[19]. Shah, L., Patel, R., Patel, S., &Maniar, J. (2014). Skew detection and correction for Gujarati printed and handwritten character using linear regression. International Journal, 4(1).
[20]. Chauhan, S., Sharma, E., &Doegar, A. (2016, September). Binarization techniques for degraded document images—A review. In 2016 5th international conference on reliability, infocom technologies and optimization (Trends and Future Directions)(ICRITO) (pp. 163-166). IEEE.
[21]. Sahu, A., & Mishra, S. (2015). Study and Analysis for Development of an Efficient OCR for Printed and Handwritten ODIA Documents: A Survey. Ijarcsse, 4(11), 14-16.
[22]. Moro, K., Fakir, M., Bouikhalene, B., El Yachi, R., & El Kessab, B. D. (2014). New approach of feature extraction method based on the raw form and his skeleton for gujarati handwritten digits using neural networks classifier.
[23]. Mukarambi, G., Dhandra, B. V., &Hangarge, M. (2012). A zone based character recognition engine for Kannada and English scripts. Procedia engineering, 38, 3292-3299.