Journal of Critical Reviews

ISSN- 2394-5125 Vol 7, Issue 14, 2020

EDGE ORIENTED FILTER AND FEATURE INVARIANT CODING FOR MULTI-LINGUAL CHARACTER SEGMENTATION AND RECOGNITION

Neerugatti Varipally Vishwanath[1], K.Manjunathachari[2], K. Satya Prasad[3],

[1]Research Scholar, Department Of ECE, JNTUK, Kakinada, Andhra Pradesh,India
[2]Professor, Department of ECE, GITAM University, Hyderabad Campus,India
[3]Rector, Vignan's Foundation for Science, Technology and Research, Vadlamudi, Guntur.

Received:11.04.2020 ***Revised: 16.05.2020*** ***Accepted: 15.06.2020***

Abstract:
In today's computer world, Automated Character Recognition has gained a significant interest in building intelligent indexing system. This paper proposes to develop automated handwritten character recognition for Multi-lingual Scripts. Total system consists of two phases, Segmentation and Recognition. In the segmentation phase, the characters are segmented from a handwritten word image and in the recognition phase, they are processed for recognition. For the segmentation phase, a new filter called, Edge Density Filter is proposed and for recognition phase, the Gabor filter is accomplished for Scale and Rotation invariant Features Extraction. *k*-Nearest Neighbor Classifier (*k*-NN) is accomplished for the recognition of character, which has less computational complexity. Simulations are conducted over three datasets among which two are publicly available and one is Self-created. Kannada and Devanagari datasets are captured from Chars74 dataset and Telugu is created voluntarily. Performance evaluation is also carried out through Recall, Precision and Accuracy and the obtained results show an outstanding performance compared to the state-of-art techniques.

Keywords: Handwritten Character Recognition, Edges, Gabor Filter, *k*-NN, Chars 74, Recall, Accuracy.

DOI: http://dx.doi.org/10.31838/jcr.07.14.106

INTRODUCTION

In recent years, text recognition from image documents has gained a significant research interest due to its widespread applicability in different fields like Signature verifications, Provision of authenticity in cheques, postal packet sorting etc. [1] In the text recognition, there is a possibility of both printed documents and handwritten documents. Compared to the handwritten image document, the character recognition in a printed image document is easy due to the uniformity of characters. The recognition of handwritten texts is a very challenging issue due to various styles of individuals in writing. Furthermore, there exists number of languages and every language has its own way of writing style. Several automatic text recognition frameworks are developed in earlier but they provide solutions only for few scripts such as English, Japanese, Chinese and Arabic etc. Furthermore, these systems are also not developed completely by considering all the issues in these scripts. Due to the variability in the writing styles, the state-of-art character recognition techniques doesnot provided satisfactory results on various types of handwriting scripts. Furthermore, most of character recognition works are developed at individual language level. Only few works are focused over the multi-lingual character recognition.

Focusing over the development of Multi-lingual character recognition system [2], [3], it can recognize multiple characters form multiple languages. Compared to a single language character recognition system, this system has more advantages and also has more challenges at the development phase. For example if we consider India, it has approximately 22 languages and also have many more unofficial languages are using for communication, and documentation. Even though some languages use same grammatical and linguistic structures, the script remains different and every script has its own style. For example, Hindi and Marathi use the same Devnagari Script and the languages Manipuri, Bangla and Assami use the Bangla Script [4]. Next, the other languages have their own script. For example, the Kannada is developed from Kadamba and Calukya Script, Telugu and Tamil are Dravidian languages and is a family of Brahmic scripts. In this way every language has its own script and developing an automatic character recognition system for every language is a complex issues. This problem is solved by Multi-lingual character recognition framework which works based on the properties of characters instead of their script [5]. Though there have been many attempts have been done in developing the Multi-lingual Character recognitions for both Indian and Non-Indian Scripts, methods that can scale across all languages and ensure reasonable results over a wide variety of image documents are not developed yet.

This paper proposed a novel Multi-lingual Character recognition framework. The proposed framework consists of two phases, first phase is for character segmentation and second phase is for character recognition. Initially, for a given printed or handwritten image document, the first phase extracts the character regions and the second phase processes the obtained character regions for recognition. A new method called edge density filter is proposed for the character segmentation based on the edge features of characters. Further the recognition framework accomplished the Gabor filter to extract the scale and orientation invariant features from character. A hybrid *k*-NN classifier which is composed of three different distance measures is accomplished for classification of input text character from the trained character samples.

Remainder of this paper is organized as follows; Section II outlines the literature survey details. Section III illustrates the detailed proposed segmentation and recognition framework. Section IV illustrates the details of simulation experiments and finally the conclusions are provided in section V.

RELATED WORK

Segmentation

Text/Character segmentation is a typical procedure in the Optical Character Recognition (OCR) Systems. Generally, a segmentation method produces a binary image and a so many character segmentation approaches are developed in the earlier [13].

Generally, segmentation is accomplished to accurately extract the characters from a word in the image document. Jia Tse et.al., [6] proposed a non-linear approach to segment the characters from grayscale image documents. Based on the general features of characters, this approach initially determined whether the characters are smeared together or not and then applied a non-linear segmentation to segment the characters. Ram Sarkar et.al., [7] proposed a novel approach for character segmentation based on a modified Run Length Smoothing Algorithm (RLSA) from Handwritten Bangla Documents. N Makhafi and O Banny [8] proposed a new scale space approach to segment the Arabic Image documents into characters based on the formation of blobs of each character. These blobs are helpful in the detection of characters in appropriate scales and also in the determination of junctions between the characters. Yi Li et.al., [9] proposed a density based estimation and level set method for segmentation of characters from an image document. Initially, this method estimates a probability map, where each element represents the probability that the current pixel is a text line. Then the level set method is applied to determine the boundary of text lines. Jewoong Ryu et.al., [10] proposed to segment the word based on the binary quadratic assignment which considers the pairwise correlations between the gaps as well as the likelihoods of individual gaps. In this approach, all the parameters are evolved through SVM structured framework.

Further, Jewoong Ryu et.al., [11] proposed to extract the text lines based on the connected components (CCs), which are used to analyze the strokes and partitions the under segmented CCs into normalized ones. Due to this normalization, the proposed method is able to predict the states of CCs for different writing style sand also for different languages. Based on these predicted estimates, a cost function is built which yields the text lines from image document. Navanet Palrecha et.al., [12] presented an algorithm for characters segmentation in Indic and roman scripts. Since the Indic Scripts are collected of characters which are connected with the Headline and two character regions may get overlapped due to matraas. This approach used Horizontal Projection profile for the extraction of headline and vertical projection for the character segmentation. Some recent works [14-16] adopted more advanced methods for the segmentation of character. In [14], Stroke Width Transform (SWT) [17] and some heuristis are accomplished to extract the background and foreground pixels. Then the segmentation is accomplished through Markov random Field (MRF) algorithm. Further in [15], the candidate regions are extracted based on extermal regions and a classifier is used for non-text regions removal based on the trained specific features. In [16], various bilateral regression models are accomplished for text extraction based on the various colors and the final results us selected based on the minimum error. Further in [18], a new Ring Radius Transform (RRT) is developed to construct the disconnected edges of a character. By integrating the medical pixels and based on the symmetrical properties of stroke of character, the inner and outer boundary is reconstructed effectively. Further in [19], Zhou et. al., proposed an inverse rendering based text segmentation framework based on the optimization of parameters in a iterative fashion. However, the obtained character regions are observed to be consisting of some background regions which have similar color properties. Next, Tian et.al., [20]considered the probabilistic statistics of stable regions to obtain a set of qualitative text regions and proposed a Multi-level MSER methodology.

Recognition

The character recognition system identifies the characters based on its features. In earlier many algorithms are developed for character recognition. Mainly the character recognition consists of three phases, namely preprocessing, feature extraction and classification. Among the three, feature extraction is most important step in recognition. According to the feature type, the earlier developed feature extraction techniques are categorized as geometrical features and texture features. The geometrical features [21] are completely dependent on the geometry of character whereas as the textures features are obtained by the transformation [22], [23].

Dash et.al., [24], [25] proposed to extract the geometrical features for handwritten character recognition from image document. A set of external symmetry axis based decomposition is accomplished in [24] to decompose the image boundary into four visual salient features such as Protrusion, Smooth Curve, Closure and Staring Segment. Based on these features, the classifier classifies the digital image using a set of classification rules. Similarly, one more geometrical feature based character recognition technique, named as Binary External Symmetry Axis Constellation (BESAC) and Fast Boolean matching character recognition technique is proposed in [25]. Two Histogram of Orientations (HOGs) are generated in this model and concatenated to BESAC feature. Further, the classification is accomplished in two phases. Further, some authors [26-29] considered the geometrical stroke features and some authors focused on the shape features [30-35]. In [28], 74 stroke classes have been identified and accomplished for Gurumukhi Script character recognition. Five different features are used with 72 different combinations of SVM and HMM for stroke classification. Further Deepthi Khanduja et.al., [29] evaluated a hybrid structural features set of the character like number of endpoints, loops, and intersection points for character identification in devanagiri script. N. R. Soora and P. S. Deshpande [30] proposed a set of Feature Vectors (FVs) based on Shape Geometry (SG). Totally two FVs are extracted here, one is based on the SG decoding of character through Triangular Area (TA) evaluation and another is based on the SG using perpendicular distance. Further M. Kumar et.al., [31] used two different features, namely parabola and curve fitting, [32] used four types of topological features, namely, Centroid features, Diagonal features, Vertically peak extent features and Horizontally peak extent features for Gurumukhi Script character recognition.

Next the texture features are generally obtained by the transformation of input character in time domain. The general transformation techniques are wavelet Transform [36], Gabor Transform [37], Scale Invariant Feature Transform (SIFT) [38], Gradient Features [39], [40] and run length features [41]. Among these features, Gabor features has a good efficiency in the provision of scale invariance and orientation invariance. Since the multi-lingual character recognition system focuses over different languages and different way of writing styles, every character needs to be represented in scale and orientation invariance format. Furthermore, compared to the geometrical features, the texture features are simple to extract and also has a significant efficiency in the recognition performance.

SEGMENTATION AND RECOGNITION FRAMEWORK

This section describes the details of proposed method for multi-lingual character recognition. The proposed method consists of two phases, (1) Character Segmentation and (2) Character Recognition. In the first phase, the all characters in a sentence or word are segmented and in the recognition phase, they are processed for recognition through machine learning algorithm.

Segmentation Framework

The segmentation framework consists of Image Preprocessing, Edge Detection and Character Segmentation through Edge Density Filter. According to the Figure.1, initially, the image document is downscaled and converted to grayscale image in the preprocessing phase. Then the grayscale image is processed for edge enhancement followed by binarization through Adaptive Thresholding. Finally the character regions are segmented through Edge Density Filter (EDF) both in vertical and horizontal directions.

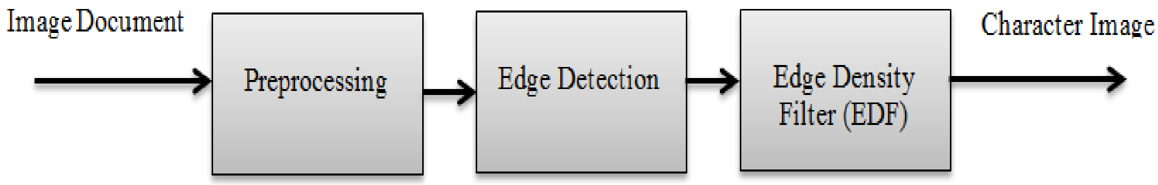


Figure.1 block diagram of segmentation scheme

Preprocessing

Generally, the script documents by hand and also the printed documents are captured at high resolution (e.g. 1024× 968). To ensure that even the small characters written in lower case are also need to be processed and has to be recognized through computer vision algorithms. Though the High Resolution images are rich in the information, they imposes a high computational cost for detecting the characters in the image document. To solve this issue, one way is to downscale the image document for the character segmentation. However the uniform downscaling results in a loss of information and leads to reduce the performance of character region segmentation. Due to this issue most of the conventional techniques not preferred the image downscaling as a part of character segmentation process.

In this paper we propose a novel image downscaling method that can reduce the size of image substantially without any decrease in the segmentation performance. It achieves a good segmentation performance when compared to the performance obtained through the original image document. Generally the height of image documents is greater than the width. For a printed document, generally the users use A4 sheet and even for handwritten document, the image captured is having greater height compared to it's width. Further the characters are also printed or written along the horizontal direction. Hence we define two different down scaling factors for horizontal and vertical directions to downscale the original image document, i.e.,

$$D_w = {}^{O_w}/_{S_w} \qquad (1)$$

$$D_h = {}^{O_h}/_{S_h} \qquad (2)$$

Where O_w and O_h denotes the width and height of original image document respectively, D_w and D_h denotes the width and height of downscaled image document respectively. Further, S_w and S_h are two different factors defined separately for width and height respectively. These two scaling factors don't have same value and the values are assigned based on the constraint, $S_h > S_w$. The large scale factor S_h is assigned to downscale the original image document along the vertical direction and S_w is assigned to downscale along the horizontal direction. Since the height of image document is greater than the width, it can be compressed more image data along vertical direction.

Edge Detection and Binarization

In this phase, the proposed method focused over the extraction of edge feature by enhancing the downscaled image followed by binarization through adaptive thresholding. In the edge detection phase, an extension of canny Edge operator is accomplished over the downscaled image for the detection of edges. Then from the obtained edges, weak edges are removed through adaptive thresholding and produce a binary edge image.

Edge Detection

In this process, we use a simple extension to the canny edge operator for edge detection. According to the figure.2, x_0 is the current pixel, and x_1, x_2, x_3 and x_4 are the neighboring pixels of x_0. Then the edge intensity e_0 for the current pixel, x_0 can be measured as,

$$e_0 = \begin{cases} \Upsilon, & if\ I_e \geq \Upsilon \\ I_e, & if\ I_e < \Upsilon \end{cases} \qquad (3)$$

Where Υ is a threshold value and I_e is an intense edge and is evaluated as

$$I_e = |x_1 + x_2 - 2x_0| + |x_3 + x_4 - 2x_0| \qquad (4)$$

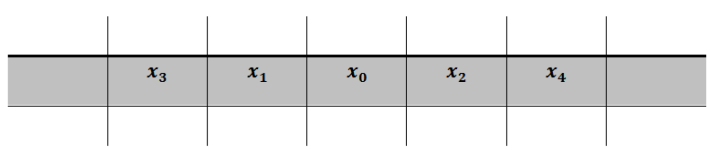


Figure.2 Intense edge for current pixel x_0

Here the main aim of edge detection is to magnify the edge in the image such that they will appear thick and for segmentation, it will become easy to recognize. Compared to the simple canny edge operator, which has a mask of size 1×3 or 3×3, this edge detection method used a single 1×5 convolution mask which is more efficient than a 1×3 mask. Further the Υ value is a run time decision, which is defined with respect to the image size. In this way, every edge pixel in the original image is obtained which constricts an edge image I_E. Some examples of edge detected images through the proposed edge detection are shown in the figure.3

(a) (b) (c)

Figure.3. Edges Detected in the one Telugu Character, (a) Original Image, (b) Canny Edge Operator, (c) Proposed Operator

Enhancement

To further enhance the obtained edge image in the earlier phase and to construct a binarized image I_B, this phase accomplished an adaptive thresholding [42] technique on the edge image I_E. For this purpose, initially the edge image I_E is divided into local segments and an average value is measured by summing up the pixel values in that segment as

$$w(x,y) = \frac{1}{w_s \times h_s} \sum_{i=(-w_s/2)}^{(w_s/2)} \sum_{i=(-h_s/2)}^{(h_s/2)} I_I(x+i, y+j) \quad (5)$$

Where $w(x,y)$ is a local window of a center pixel located at (x,y). Next, w_s is the width of local segment and h_s is the height of local segment and I_I is an integral image obtained by summing up the all pixels in the grayscale Edge image, I_E. Then the binary edge image I_B is obtained by thresholding every pixel in the integral image I_I based on the adaptive threshold which is computed from local segment in I_I. The process of binary edge image I_B is obtained as

$$I_B(x,y) = \begin{cases} 255, & if\ I_I(x,y) \geq \beta w(x,y) \\ 0, & if\ I_I(x,y) < \beta w(x,y) \end{cases} \quad (6)$$

Where β is an arbitrary constant and its value is fixed to 0.7. Here the size of local segment is set to 7×5.

Edge Density Filter (EDF)

This filter is applied over the binary edge image I_B to extract the character regions. Based on the connectivity of pixels in every character region, this filter has achieved efficient results in the segmentation of characters. The main reasons behind the consideration of [43] are as follows;

1. Characters printed/handwritten, generally exhibits a high edge density.
2. For a printed document image, the characters are printed along horizontal direction and each character is of same size. Similarly for handwritten image document, the characters are written along the same horizontal direction but the size is not exactly same. It is approximately same.

The main working methodology of the EDF is by observing the above mentioned clues. The EDF connects the regions with high edge density and removes the other regions from the binary edge image I_B. A simple illustration about the EDF is depicted in the figure.4, below.

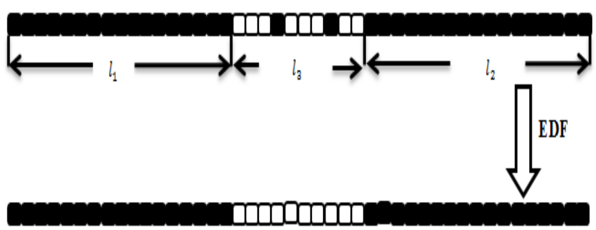


Figure.4. EDF Accomplishment

According to figure.4, the black dot spots represent the non-edge pixels and white dot spots represent the edge pixels. For instance, let l_1 and l_2 are the two continuous black line segments of lengths l_{b1} and l_{b2} respectively. Our main aim is to quantify the target line segment l_3 based on the number of white spots and number of black spots. Let w_l be the number of white spots in the target line segment and b_l be the number of black spots in the target line segment, the decision is taken based on the l_d and is measured as

$$l_d = {w_l}/{w_l + b_l} \quad (7)$$

Further, for any printed/handwritten document, there is a chance to observe a non-character region on the both sides, which can be a long black line segments l_1 and l_2. Once the character regions are initialized in the image document, we can observe a non-lengthy black line segments. Here the length map is obtained by accumulating the continuous white and black pixels. Here the black pixel is counted as 1 and if a continuous black pixel is observed, a cumulative length is measured by grouping them. Similarly, the white pixel is counted as -1 and if continuous white pixels are observed, a cumulative length is measured by grouping them but in negative. For example, if we observe 12 continuous black pixels are observed then the cumulative length map is 12 whereas for continuous white pixels, it is -12. This process obtains only the character regions in the horizontal direction only. To obtain the characters in the next line, the vertical scanning is applied based on the obtained line length maps for every character in the binary image, the proposed EDF is applied to obtain the character regions. It is based on the parameters such as binary edge image I_B; minimum length threshold for black line T_{min}, for l_1 and l_2; the gap length threshold T_{gap}, for line segment l_3; and the line density threshold T_{dl}, for l_3. For a given binary image I_B, for each line segment, if we found two line segments l_1 and l_2 of lengths l_{b1} and l_{b2} respectively, if $l_{b1} > T_{min}$ and $l_{b2} > T_{min}$, then calculate l_d using Eq.(7) for l_3 and compare it with the line density threshold

T_{dl}. If it is greater than the T_{dl}, set all the pixels of l_3 to white in I_B, otherwise set them to black.

In this manner, the characters regions are segmented from the image document and then processed for recognition framework.

Recognition Framework

Once the character regions are extracted from image document, then they are processed for recognition. The proposed recognition framework consists of Preprocessing, Feature Extraction and classification of characters. In the initial phase, the image is subjected to cropping of the background regions from all sides which results in a image starting with text pixel from all sides. Further the feature extraction phase extracts the required features from character and then the classifier classifies the character based on the present knowledge given in training phase.

Feature Extraction

Once the pre-processing phase is completed, the grayscale character image is processed for feature extraction. In this phase, the proposed method accomplished Gabor Filter for feature extraction. Since the handwritten characters are of different size and also having different orientations, representing them with a scale and orientation invariant features is important and the Gabor filter is one of the best filter which complete this responsibility effectively.

Towards features extraction, this work accomplished the Gabor Filter due to its effectiveness in the feature extraction in different orientations [44]. Since the Gabor filter extracts the features those are scale-invariant and orientation-invariant, this paper considered it for Orientational features extraction. Here the Gabor filter is accomplished in different scales such as 3×3 , 5×5, 7×7, 9×9, and 11×11, and eight orientations such as 0^0, 45^0, 90^0, 135^0, 180^0, 225^0, 270^0, and 315^0. So totally for each frame, we will get $5 \times 8 = 40$ feature maps. The main mathematical formula for Gabor filter is shown as

$$G(x,y) = exp\left(\frac{X^2+\gamma Y^2}{2\sigma^2}\right) \cos\left(\frac{2\pi}{\lambda}x\right) \quad (8)$$

Where

$$X = xcos\theta - ysin\theta, \;\; Y = xsin\theta + ycos\theta \quad (9)$$

Where (x, y) is position relative to the center of filter. A simple representation of Gabor Filter accomplishment is depicted in the following figure.5, below.

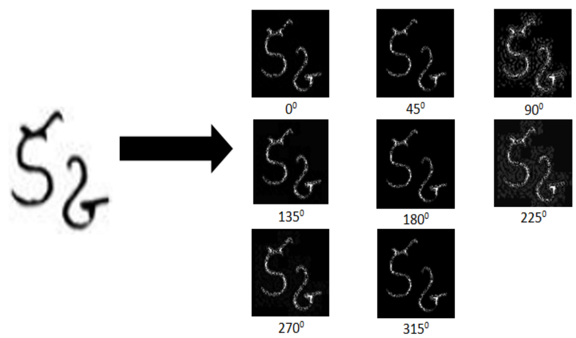


Figure.5 Gabor Filter in Eight Orientations

However the obtained 40 feature maps are high in number and create much computational burden. Hence, to reduce this computational burden, here a max pooling [45] is applied over the 40 feature maps of every frame. In other words, we will pick a maximum value from all feature maps with filter scale in each orientation. The max pooling in different scales is performed as

$$F_{max} = \max_{(x,y,\theta_s)} [F_{3\times3}(x, y, \theta_s), \dots\dots\dots, F_{11\times11}(x, y, \theta_s)] \quad (10)$$

Where F_{max} is the maximum feature map obtained through the max-pooling, $F_{k\times k}(x, y, \theta_s)$ is the feature map at $k \times k$ scale and at θ_s orientation. In this manner, we will get totally eight feature maps, one from each orientation.

Character Classification

Upon the feature extraction from the character region, they are formulated into a 1-D vector and then processed for classification. Here the classification phase uses the most popular k-NN classifier [46]. In this k-NN classifier, k-neighbors are obtained by performing the comparison between the test character features and trained characters features. The k-NN classifier accuracy in influenced by two factors, (1) distance metric accomplished for the similarity evaluation between testing and training features, and (2) the value of k. Here three different distance measures are considered for similarity evaluation. They are Cosine Distance, City block distance, and Euclidean Distance. Let's x be the feature vector of an input character, and feature vector of trained characters is $Y_i (i = 1,2 \dots. N)$, where N is the total number of characters trained, the three distances are measured as;

$$D_{\cos}(x, y) = \frac{\sum_{i=1}^{P} x_i y_i}{\sqrt{\sum_{i=1}^{P} x_i^2} \sqrt{\sum_{i=1}^{P} y_i^2}} \quad (11)$$

Where $D_{\cos}(x, y)$ is the cosine distance between feature vector of an input character, and feature vector of trained character $y \in Y$.

$$D_{\mathrm{cb}}(x, y) = \sum_{i=1}^{P} |x_i - y_i| \quad (12)$$

Where $D_{\mathrm{cb}}(x, y)$ is the city block distance between feature vector of an input character, and feature vector of trained character $y \in Y$.

$$D_{\mathrm{ed}}(x, y) = \sqrt{\sum_{i=1}^{P} (x_i - y_i)^2} \quad (13)$$

Where $ed(x, y)$ is the Euclidean distance between feature vector of an input character, and feature vector of trained character $y \in Y$.

Here the P denotes the dimensions of feature vectors. At every instant, the first k-nearest neighbors are considered and this process is repeated for multiple runs and based on the obtained neighbors, the input character is classified into one of the characters which are in database.

SIMULATION RESULTS

To verify the proposed system, simulations are conducted over different multi-lingual datasets. The present simulation considered three datasets, two publicly available datasets and one self-created dataset. A 10-fold cross validation is executed every dataset and the accuracy is measured by averaging the obtained values at each validation.

Databases Used

The two publicly available datasets used here for summation are Kannada Dataset and Devanagari Dataset, and the self-created is Telugu Dataset. The Kannada Dataset is acquired from the most popular Chars74 dataset [47]. Both the Handwritten and Printed characters of Kannada Script can be observed in the Chars74 dataset. Totally there are more than 657 classes of characters of Kannada script are available here. Every character class is represented in 25 different styles with different scales, orientations and thickness. All the characters are in image (.PNG) format. Some sample handwritten character images from Kannada Script is shown in the below figure.

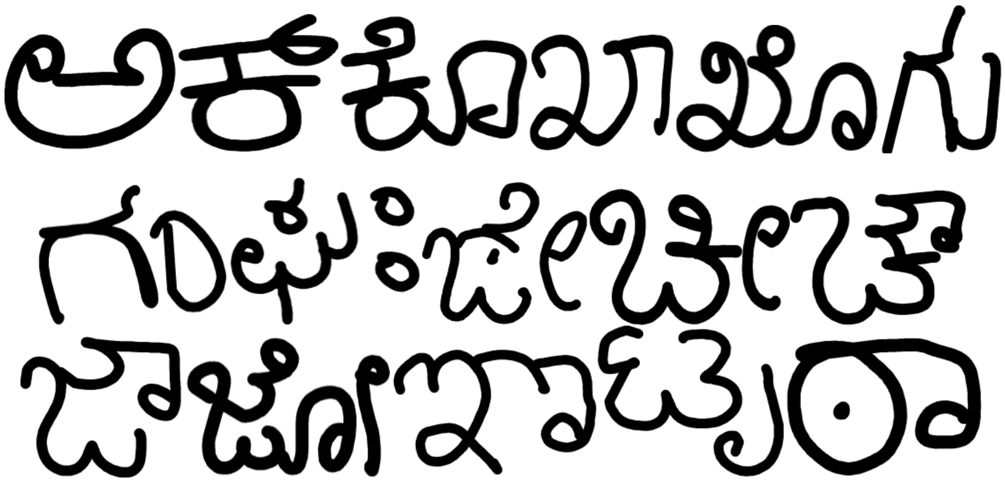


Figure.6 Sample Handwritten character images from Kannada Dataset

The next publicly available dataset, Devanagari Dataset is acquired from UCI machine learning repository [48]. This data set consists of Handwritten Devanagari Character images. There are totally 46 classes of characters with 2000 samples for each class. Under every class, 1700 samples are considered for training and 300 samples are considered for testing. All the character images are in image (.PNG) format. The size of every character image is 32×32. The actual size of character is 28×28 but 2 pixels are added on all four sides of actual character. Some sample handwritten character images from Devanagari Script is shown in the below figure.

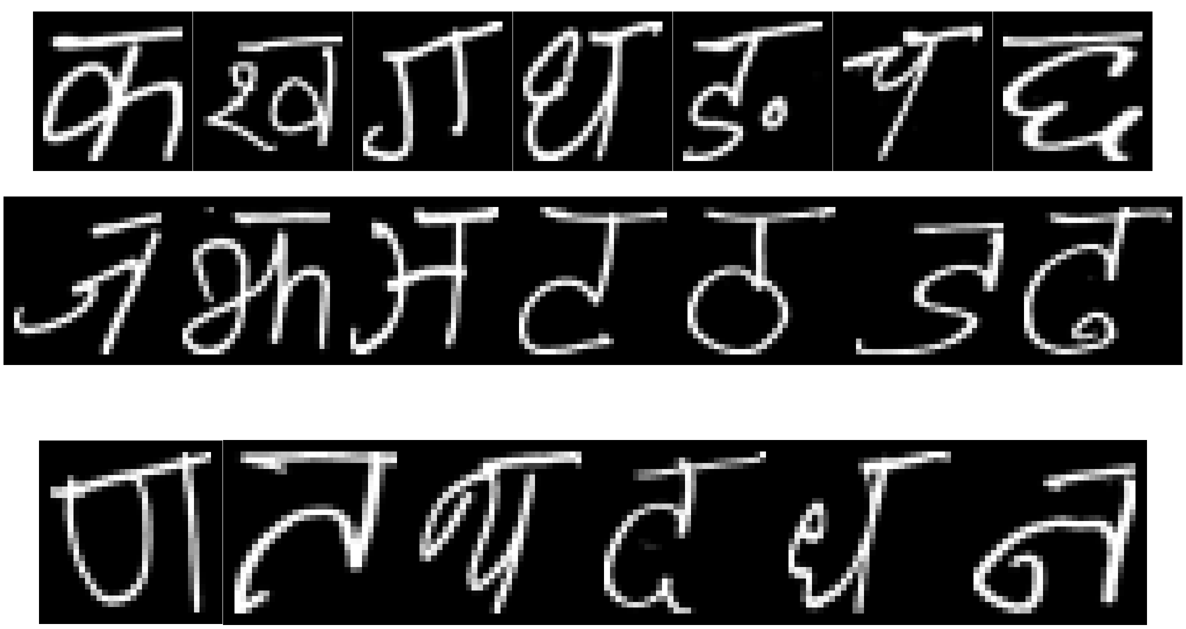


Figure.7 Sample Handwritten character images from Devanagari Dataset

Next, the simulation experiments considered the Self-Created Telugu Dataset. This dataset is created voluntarily through by asking 25 people who have different ways of writing style. For each person, totally 52 characters (16-Achulu and 36-Hallulu) are asked to write in his/her own style. Further the same people are asked to write some words in Telugu. These words are used for testing purpose. Sample character images of this self-created dataset are shown below.

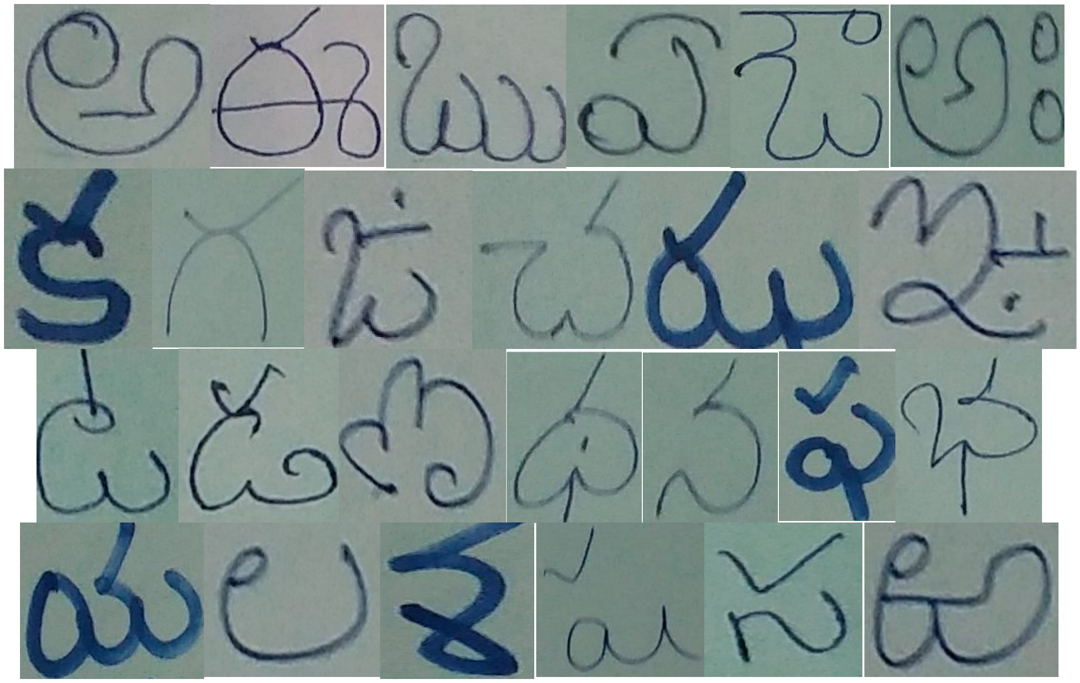


Figure.8 Sample Handwritten character images from Telugu Dataset

Results
In this section, the segmented results are stipulated. Testing is done for all the languages by giving a handwritten word as an input for the segmentation phase. Input handwritten words and their segmented results are shown below;

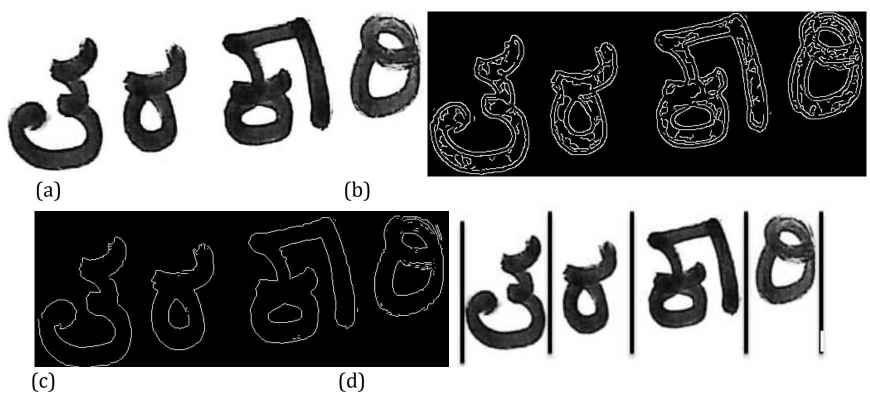


Figure.9 (a) Test Kannada Word Image, (b) Edge Detected image, (c) Binarized and Enhanced Image, (d) Segmented Image

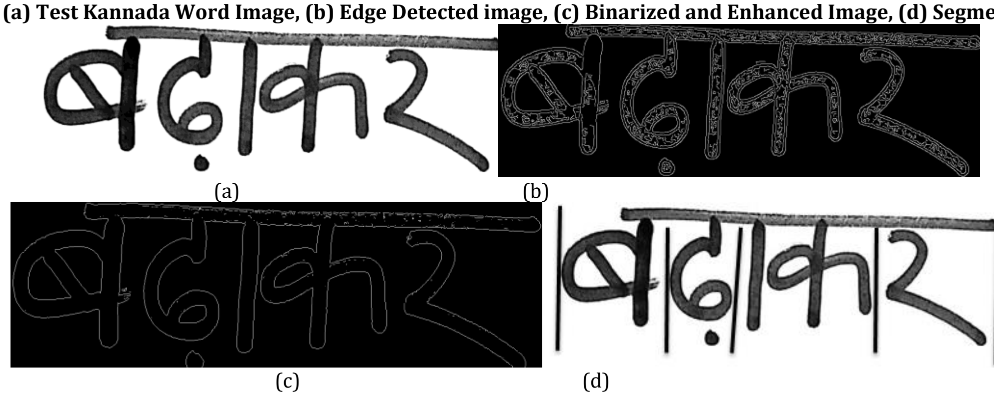
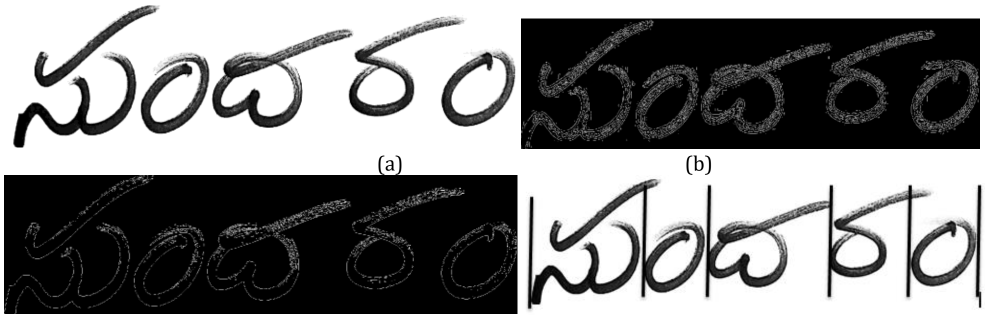


Figure.10 (a) Test Devanagari Word Image, (b) Edge Detected image, (c) Binarized and Enhanced Image, (d) Segmented Image



(a) (b) (c) (d)

Figure.11 (a) Test TeluguWord Image, (b) Edge Detected image, (c) Binarized and Enhanced Image, (d) Segmented Image

Figure.9 shows the obtained results after the accomplishment of proposed segmentation algorithm over the Kannada word image. Figure.9(a) is the original handwritten gray-level image and 9(b) shows the obtained image after Edge Detection. After obtaining the edge image, to remove the unnecessary edges, the binarization is applied followed edge enhancement and the obtained result is shown in figure.9(c). Finally after the accomplishment of EDF over the edge enhanced image, the characters are segmented based on their lengths and widths. Since the proposed EDF applies a simultaneous vertical and horizontal scanning over the image, a character with length 1's is segmented as one character. Further a gap between the characters is represented by 0's which is considered as background. By counting the total pixel count which have a continuous 1's and 0's, the characters are segmented and the resultant is shown 9n figure.9(d). Similarly, the segmented results of Devanagari and Telugu word images are shown in figure.10 and figure.11 respectively.

Performance Evaluation

Under this section, the performance is measured with respect to different performance metrics. These metric are measured according to the classified results. For this purpose, initially, the system is trained with all the characters as discussed in the Section IV.A. After training the characters, the testing is accomplished over handwritten characters. Tenfold cross-validation is used for testing results using 70% of the data as training and 30% for validation. Further, testing is also done over the characters obtained after the segmentation.

In this paper, the performance metrics namely Accuracy, Precision, Detection Rate or Recall, False Negative Rate (FNR) and F-Score are considered to evaluate the performance of proposed approach. After testing, the obtained results are formulated into a confusion matrix. Depends on the obtained numerical results, the True Positives (TPs), True Negatives (TNs), False Positives (FPs), and False Negatives (FNs) are measured. Based on the obtained TP, TN, FP and FN values from the confusion matrix, performance metrics are evaluated and the respective mathematical representation is given as;

$$Recall = \frac{TP}{TP+FN} \quad (14)$$

$$Precision = \frac{TP}{TP+FP} \quad (15)$$

$$F-Score = \frac{2\times Recall\times Precision}{Recall+Precision} \quad (16)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (17)$$

$$False\ Negative\ Rate = \frac{FN}{TP+FN} \quad (18)$$

Table.1 Performance Evaluation over handwritten Characters

Script	Recall (%)	Precision (%)	F-Score (%)	FNR (%)
Kannada	77.5602	79.0027	78.2748	22.4398
Devanagari	78.8928	80.3353	79.6075	21.1072
Telugu	75.4747	76.9172	76.1891	24.5253

Table.2 Performance Evaluation over Segmented Characters

Script	Recall (%)	Precision (%)	F-Score (%)	FNR (%)
Kannada	75.6025	77.1536	76.3702	24.3975
Devanagari	76.9317	78.4862	77.7012	23.0683
Telugu	73.8510	75.0681	74.4546	26.1490

Table.3 Performance Comparison

Method	Database	Feature Type	Classifier	Recall (DR) (%)
Shi et.al., [21]	Chars74	Stroke Detectors	SVM	71.8
Tian et.al., [40]	Chars74	HoG	CNN	76.6
Proposed	Chars74	Gabor Features	k-NN	77.6

Table.1 reveals the details of performance metrics obtained after the accomplishment of proposed approach over the all dataset characters that are written in different styles. For every dataset, the Performance metrics are measured after obtaining the classification results, i.e., based on the output obtained at k-NN classifier. The metric, recall denotes the total number of samples classified as correct for a given total number of input samples. Similarly the precision is measured as total number of samples classified as correct for a total present samples. Further the F-Score is a harmonic mean of Recall and Precision. The FNR is an opposite version of Recall, i.e., 100-Recall=FNR. In the table.1 the obtained results are after testing the characters directly through the proposed recognition framework and the results in table.2 are obtained for the recognition of characters which are segmented from an input word image. As it can be observed from the two tables, the results of table.2 are less compared to table.1 because the characters in the word are having so many constraints such as Overlapping, Cursive and Improper Background etc.

Next, table.3 shows the comparative analysis between the proposed and conventional approach through recall. It can be observed from teable.3, the recall of proposed approach is high compared to the conventional approaches. It is due to the accomplishment of EDF at the segmentation phase and Gabor filter for feature extraction. Further the k-NN is also not formulated through a simple Euclidean distance. It is formulated through three different distances, which had shown a good contribution in recognizing the perfect character.

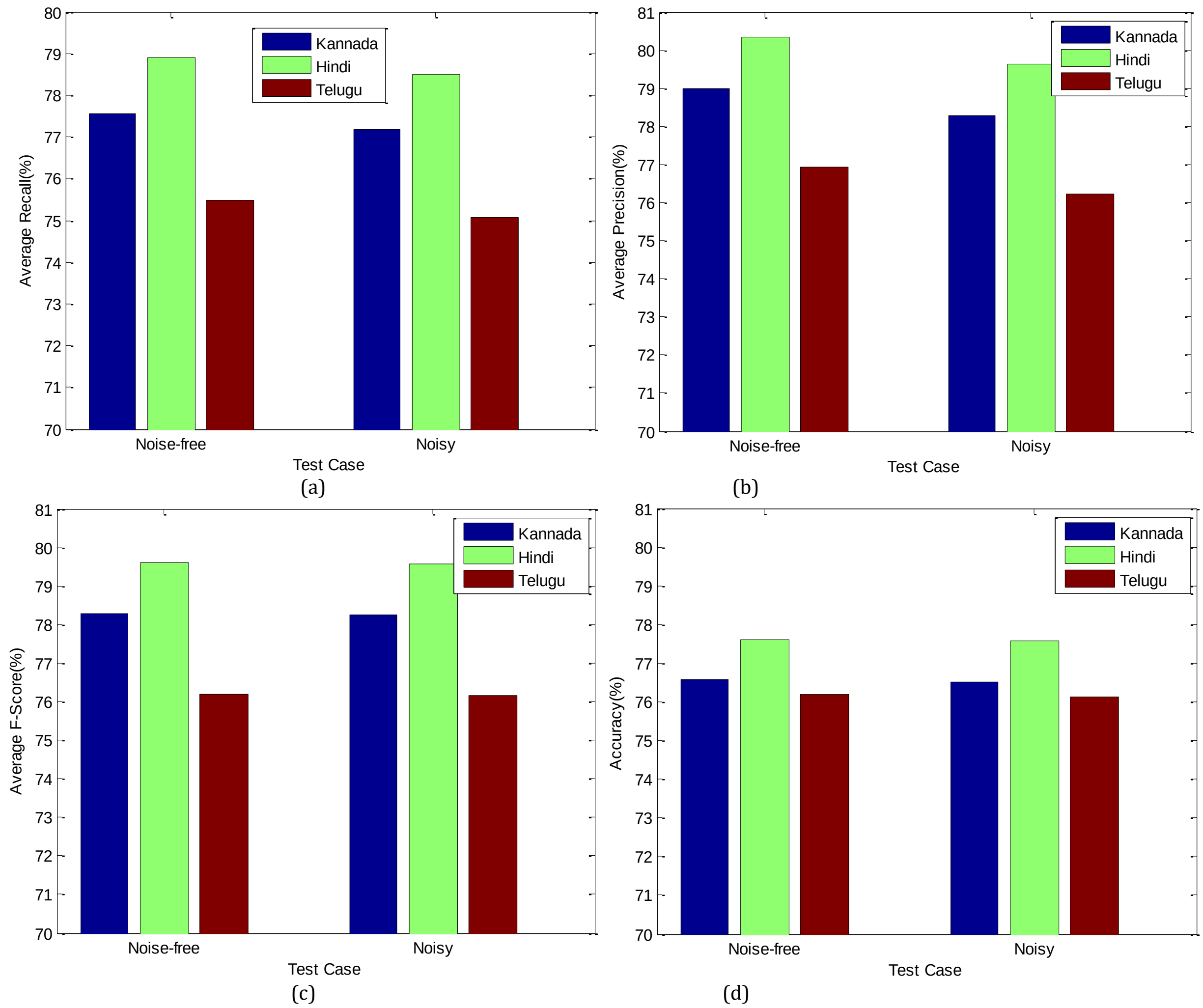


Figure.12 Perofmrance analysis through (a) Average Recall, (b) Avergae Precision, (c) Average F-Score and (d) Accuracy

To further alleviate the performance of developed mechanism, simulation experiments are conducted in the presence of noise. Figure.12 shows the results obtained under noise-free and noisy situations for all the test languages. Since the noise is an external factor which creates more confusion, the proposed approach attained a lesser performance in the noisy case compared to the noise-free case. Even though it is less, the overall Recall of proposed approach is high compared to the conventional approaches. On an average, the proposed approach obtained a recall rate as 77.6% whereas it is of 76.6 % and 71.8% for conventional approaches, Tian et. al., [40] and Shi et. al., [21] respectively.

CONCLUSION

This paper proposed a novel Multi-lingual Character recognition framework. The total framework is accomplished in two phases, first phase is for character segmentation and second phase is for character recognition. The proposed new technique, EDF at segmentation phase and Gabor Filter Feature extraction at recognition phase had shown a greater significance in the segmentation and recognition of handwritten characters even under abnormal environments such as Noise and Word image with overlapping. Simulations are carried out over the proposed system through three different datasets and the performance is measured through the Performance metric such as Recall, Precision, and Accuracy. Results had shown that the proposed approach outperforms the conventional approaches.

REFERENCES

1. Vinciarelli, A.: 'A survey on off-line cursive word recognition', *Pattern Recognit.*, 2002, 35, (7), pp. 1433–1446.

2. V. Singhal, N. Navin, and D. Ghosh, "Script-based classification of hand-written text documents in a multilingual environment," in *Proc. IWRIDEMIM*, Mar. 2003, pp. 47-54.
3. L. Peng, C. Liu, X. Ding, and H. Wang, "Multilingual document recognition research and its application in China," in *Proc. DIAL*, 2006, pp. 126-132.
4. Minesh Mathew, Ajeet Kumar Singh and C. V. Jawahar, "Multilingual OCR for Indic Scripts", 2016 12th IAPR Workshop on Document Analysis Systems (DAS), Santorini, Greece.
5. D. Arya, T. Patnaik, S. Chaudhury, C. V. Jawahar, B. B. Chaudhuri, A. G. Ramakrishna, C. Bhagvati, and G. S. Lehal, " Experiences of Integration and Performance Testing of Multilingual OCR for Printed Indian Scripts," in J-MOCR Workshop, ICDAR, 2011.
6. Jia Tse, Christopher Jones, and Dean Curtis, "An OCR-independent character segmentation using shortest path in grayscale document images," Sixth International Conference on Machine Learning and Applications (ICMLA), California, OH, USA, 2007.
7. Ram Sarkar, Samir Malakar, Nibaran Das, Subhadip Basu, Mahantapas Kundu, Mita Nasipuri, "Word Extraction and Character Segmentation from Text Lines of Unconstrained Handwritten Bangla Document Images", Journal of Intelligent Systmes, Vol.20, Issue 3, pp.227-260, 2011.
8. N Makhafi and O Banny, "Scale-Space Approach for Character Segmentation in Scanned Images Of Arabic Documents," Journal of Theoretical and Applied Information Technology, Vol. 94. No.2, 2016. pp. 444-450.
9. Yi Li, Yefeng Zheng, David Doermann, "Script-Independent Text Line Segmentation in Freestyle Handwritten Documents", IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 30 , Issue: 8, Aug. 2008).
10. Jewoong Ryu, Hyung ii Koo, Nam lk cho, "Word Segmentation Method for Handwritten Documents based on Structured Learning", IEEE Signal Processing Letters, Volume: 22 , Issue: 8 , Aug. 2015.
11. Jewoong Ryu, Hyung ii Koo, Nam lk cho, "Language-Independent Text-Line Extraction Algorithm for Handwritten Documents", IEEE Signal Processing Letters, Volume: 21, Issue: 9, Sept. 2014.
12. Navanet Palrecha, Anvaya Rai, Ajay Kumar, Shkha Srivasthava, VIpin Tyagi, "Character segmentation for multi lingual Indic and Roman scripts", 2011 IEEE 7th International Colloquium on Signal Processing and its Applications, Penanag, Malaysia.
13. P. Stathis, E. Kavallieratou, N. Papamarkos, An evaluation survey of binarization algorithms on historical documents, in: International Conference on Pattern Recognition(ICPR), 2008, pp.1–4.
14. A. Mishra, K. Alahari, C. Jawahar, An MRF model for binarization of natural scene text, in: International Conference on Document Analysis and Recognition (ICDAR), 2011, pp.11–16.
15. L. Neumann, J. Matas, Real-time scene text localization and recognition, in: Computer Vision and Pattern Recognition (CVPR), 2012, pp.3538–3545.
16. J. L. Feild, E. G. Learned-Miller, Improving open-vocabulary scene text recognition, in: International Conference on Document Analysis and Recognition (ICDAR), 2013, pp.604–608.
17. B. Epshtein, E. Ofek, Y. Wexler, Detecting text in natural scenes with stroke width transform, in: Computer Vision and Pattern Recognition (CVPR), 2010, pp. 2963–2970.
18. P. Shivakumara, T. Q. Phan, S. Bhowmick, C. L. Tan, U. Pal, A novel ring radius transform for video character reconstruction, *Pattern Recognit*. Vol.46, 2013, pp.131– 140.
19. Y. Zhou, J. Feild, E. Learned-Miller, R. Wang, Scene text segmentation via inverse rendering, in: International Conference on Document Analysis and Recognition (ICDAR), 2013, pp.457–461.
20. S. Tian, S. Lu, B. Su, C. L. Tan, Scene text segmentation with multi-level maximally stable extremal regions, in: International Conference on Pattern Recognition (ICPR), 2014, pp.2703–2708.
21. C. Z. Shi, S. Gao, M.-T. Liu, C. Z. Qi, C. H. Wang, and B. H. Xiao, ``Stroke detector and structure based models for character recognition: A comparative study,'' *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4952-4964, Dec. 2015.
22. V. N. M. Aradhya, G. H. Kumar, and S. Noushath, "Multilingual OCR system for South Indian scripts and English documents: An approach based on Fourier transform and principal component analysis," *Eng. Appl. Artif. Intell.*, vol. 21, no. 4, pp. 658-668, Jun. 2008.
23. P. Sahare and S. B. Dhok, "Script identification algorithms: A survey," *Int. J. Multimedia Inf. Retr.*, vol. 6, no. 3, pp. 211-232, Sep. 2017.
24. K. S. Dash, N. B. Puhan, and G. Panda, "Unconstrained handwritten digit recognition using perceptual shape primitives," in *Pattern Analysis and Applications*. London, U.K.: Springer, Nov. 2016, pp. 1-24.
25. K. S. Dash, N. B. Puhan, and G. Panda, "BESAC: Binary external symmetry axis constellation for unconstrained handwritten character recognition," *Pattern Recognit. Lett.*, vol. 83, no. 3, pp. 413-422, Nov. 2016.
26. B. Zhu and M. Nakagawa, "A robust method for coarse classifier construction from a large number of basic recognizers for on-line handwritten Chinese/Japanese character recognition," *Pattern Recognit.*, vol. 47, no. 2, pp. 685-693, Feb. 2014.
27. N. Das, R. Sarkar, S. Basu, P. K. Saha, M. Kundu, and M. Nasipuri, "Handwritten *Bangla* character recognition using a soft computing paradigm embedded in two pass approach," *Pattern Recognit.*, vol. 48, no. 6, pp. 2054-2071, Jun. 2015.
28. K. Verma and R. K. Sharma, "Comparison of HMM- and SVM-based stroke classifiers for Gurmukhi script," *Neural Comput. Appl.*, vol. 28, pp. 51-63, Dec. 2016.
29. D. Khanduja, N. Nain, and S. Panwar, "A hybrid feature extraction algorithm for devanagari script," *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, vol. 15, no. 1, pp. 2:1-2:10, Jan. 2016.
30. N. R. Soora and P. S. Deshpande, "Novel geometrical shape feature extraction techniques for multilingual character recognition," *IETE Tech. Rev.*, vol. 34, no. 6, pp. 612-621, Oct. 2016.
31. M. Kumar, R. K. Sharma, and M. K. Jindal, "Efficient feature extraction techniques for off-line handwritten Gurmukhi character recognition,'' *Nat. Acad. Sci. Lett.*, vol. 37, no. 4, pp. 381-391, Aug. 2014.
32. M. Kumar, M. K. Jindal, and R. K. Sharma, "A novel hierarchical technique for off-line handwritten Gurmukhi character recognition," *Nat. Acad. Sci. Lett.*, vol. 37, no. 6, pp. 567-572, Dec. 2014.
33. M. Kumar, R. K. Sharma, and M. K. Jindal, "A novel feature extraction technique for off-line handwritten *Gurmukhi* character recognition," *IETE J. Res.*, vol. 59, no. 6, pp. 687-691, Sep. 2014.
34. S. Bag, G. Harit, and P. Bhowmick, "Recognition of Bangla compound characters using structural decomposition," *Pattern Recognit.*, vol. 47, no. 3, pp. 1187-1201, Mar. 2014.
35. N. R. Soora and P. S. Deshpande, ``Robust feature extraction technique for license plate characters recognition,'' *IETE J. Res.*, vol. 61, no. 1, pp. 72-79, Jan. 2015.

36. R. Kumar and K. K. Ravulakollu, “Off-line handwritten Devnagari digit recognition,” *ARPN J. Eng. Appl. Sci.*, vol. 9, no. 2, pp. 109_115, Feb. 2014.
37. J. R. Prasad and U. Kulkarni, “Gujrati character recognition using weighted *k*-NN and Mean *X*2 distance measure,” *Int. J. Mach. Learn. Cybern.*, vol. 6, no. 1, pp. 69-82, Feb. 2015.
38. O. Surinta, M. F. Karaaba, L. R. B. Schomaker, and M. A. Wiering, “Recognition of handwritten characters using local gradient feature descriptors,” *Eng. Appl. Artif. Intell.*, vol. 45, pp. 405-414, Oct. 2015.
39. P. Singh, A. Verma, and N. S. Chaudhari, “Feature selection based classifier combination approach for handwritten Devanagari numeral recognition,” *Sadhana*, vol. 40, no. 6, pp. 1701-1714, Sep. 2015.
40. S. Tian *et al.*, “Multilingual scene character recognition with co-occurrence of histogram of oriented gradients,” *Pattern Recognit.*, vol. 51, pp. 125-134, Mar. 2016.
41. S. Raja and M. John, “A novel Tamil character recognition using decision tree classifier,” *IETE J. Res.*, vol. 59, no. 5, pp. 569-575, Sep. 2014.
42. D. Bradley, G. Roth, “Adaptive thresholding using the integral image,” *J. Graph. Tools*, vol.12, no.2, pp.13-21, (2007).
43. Yule Yuan, Wenbin Zou, Yong Zhao, Xinan Wang, Xuefeng Hu, and Nikos Komodakis, “A Robust and Efficient Approach to License Plate Detection”, IEEE Transactions on Image Processing, 2016.
44. Li Liu, Ling Shao, *Senior Member, IEEE*, Xiantong Zhen, and Xuelong Li, “Learning Discriminative Key Poses for Action Recognition”, IEEE Transactions on Cybernetics, Vol. 43, No. 6, December 2013
45. M. Riesenhuber and T. Poggio, “Hierarchical models of object recognition in cortex,” *Nat. Neurosci.*, vol. 2, pp. 1019–1025, 1999.
46. Parul Sahare And Sanjay B. Dhok, “Multilingual Character Segmentation and Recognition Schemes for Indian Document Images”, *IEEE Access*, 2018. pp.10603-10617.
47. T. E. de Campos, B. R. Babu and M. Varma, “Character recognition in natural images”, In *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal*, February 2009.
48. S. Acharya, A.K. Pant and P.K. Gyawali, “Deep Learning Based Large Scale Handwritten Devanagari Character Recognition”, *In Proceedings of the 9th International Conference on Software, Knowledge, Information Management and Applications* (SKIMA), pp. 121-126, 2015.