 doi:10.20944/preprints202006.0048.v1

Brahmi word recognition by supervised techniques

Neha Gautam [1*], Soo See Chai[1], Sadia Afrin[2], Jais Jose[3]

[1] Faculty of Computer Science and Information Technology, University Malaysia Sarawak
[2] Institute of Cognitive Science, Universität Osnabrück
[3] Amity University, Noida, India
[*1] nehagautam1208@gmail.com
[2] sadiaaaffrin@gmail.com

Abstract: Significant progress has made in pattern recognition technology. However, one obstacle that has not yet overcome is the recognition of words in the Brahmi script, specifically the recognition of characters, compound characters, and word because of complex structure. For this kind of complex pattern recognition problem, it is always difficult to decide which feature extraction and classifier would be the best choice. Moreover, it is also true that different feature extraction and classifiers offer complementary information about the patterns to be classified. Therefore, combining feature extraction and classifiers, in an intelligent way, can be beneficial compared to using any single feature extraction. This study proposed the combination of HOG +zonal density with SVM to recognize the Brahmi words. Keeping these facts in mind, in this paper, information provided by structural and statistical based features are combined using SVM classifier for script recognition (word-level) purpose from the Brahmi words images. Brahmi word dataset contains 6,475 and 536 images of Brahmi words of 170 classes for the training and testing, respectively, and the database is made freely available. The word samples from the mentioned database are classified based on the confidence scores provided by support vector machine (SVM) classifier while HOG and zonal density use to extract the features of Brahmi words. Maximum accuracy suggested by system is 95.17% which is better than previously suggested studies.

1. Introduction

Various type of issues have been solved by using machine learning [1] and a recent problem in pattern recognition is automatic word recognition, which can be solved by an optical character recognition (OCR) system. OCR system can help to automatically able to convert the several kinds of documents like scanned papers, digital pictures took into editable and searchable data [2]. Various works already completed related to English and Roman script recognition compare to South and Central Asian scripts because South and Central Asian scripts are more cursive compare to English and Roman script [3]. Recognition of South and Central Asian scripts become extra challenging task when study is focused on ancient scripts recognition because of the significant variations in the structure of the characters, lack of resources, controversy statements, etc.

Brahmi script belongs to one of the ancient writing scripts applied in South, Southeast, and East Asia. A study identified 198 different current scripts example: Devanagari, Bangla, Gurmukhi, Gujarati, Oriya, etc. which originated from the Brahmi script [4]. So, it can say that the Brahmi script has a wealthy background.

In the work of Brahmi scripts recognition, the suitable features of the Brahmi scripts were extracted and after that, the extracted features were classified by using the suitable classifier. For instance, geometric features [4], zoning method [5, 6] was applied to obtain the features of the Brahmi scripts and extracted features were classified by some set of rules [4], template matching [5] and Artificial Neural Network (ANN) [6].

Some works have been found out to recognize the Brahmi script, but the performance of the existing system can be increased because exist works did not focus on finding out the best feature extraction techniques and classifier for achieving the highest accuracy. Whereas, the correct feature extraction and classifier is significant to increase the recognition rate of any recognition system [7]. The accuracy of the recognition system is changed while the different feature extraction methods are applied because the suitable feature extraction technique depends on the feature of the script. It also can see that one feature extraction method does not receive the same accuracy of two different script recognition.

Thus, the accuracy of the Brahmi word recognition system can be increased by the feature extraction methods and classifiers. So, the focus point of this study, to focused on the recognition rate of various feature extraction techniques and classifiers to recognize the Brahmi words to discover the best combination of feature extraction and classifier for Brahmi word recognition system.

Characteristic and properties of the Brahmi script is explained in section II. Details of the earlier methods and frameworks related to this study are presented in part III. details of the Brahmi word dataset are discussed in section IV and the entire recognition process of Brahmi word recognition is mentioned in section V. Results and comparison with previous systems are delivered in part VI and finally concluding remarks is mentioned in chapter VII.

2. Brahmi Script

Brahmi script is an ancient script and the period of Brahmi script was 3rd BCE to 6th centuries (CE) [4]. This script was used to write various religious books (Buddhist and Jainism). At one time, Brahmi script was referred as the "pin-man" script that is "stick figure" script in English. In other words, Brahmi characters have geometric features like lines, curves, corners, and dots [4] and these features

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2020 doi:10.20944/preprints202006.0048.v1

can help to recognize the Brahmi script from the various scripts (figure 1).

2.1. Properties of Brahmi script

2.1.1 Compound character are the modified characters in shape, which is the combination of consonant and vowel. The modifications in shape can see on the left, right, top, or bottom of the consonant, it depends on the vowel [8]. Some time, two vowels follow a consonant and become a new character that called complex compound characters. Both properties follow by Brahmi script like Devanagari script, Bangla script.

2.1.2 Brahmi script has 368 total numbers of characters including the number of consonants are 33, vowels are 10, and the rest (325) are compound characters [9].

2.1.3 Left side to right follow to write text in Brahmi script.

2.2. Characteristic

Brahmi consonants are followed by different vowels to make compound characters. The features are added in the consonants to create compound characters called "Matra". Mostly, these "Matra" can be seen on the outer edge of the consonants (but not always because it also depends on the shape of the consonants) or using a dot feature (.) that is added after the consonant. The first character is the consonant, and the rest are the compound characters. Nine types of features ("Marta") are used to make compound characters.

Figure 1: Character and vowels of Brahmi script [10]

3. Literature Review

Of the previous works that have developed Brahmi script recognition systems, only a few focused on Brahmi character recognition and even fewer reported working on Brahmi word recognition. The history of Brahmi script recognition can be traced back to 1983. In this year, Siromoney, et al. [9] developed a system to recognize all Brahmi characters, including all 368 standard vowels, consonants, and compound characters. Their work, however, ignored all complex compound characters. A coded run method was used for the recognition of machine-printed characters of the Brahmi script, with each Brahmi character manually changed into a rectangular binary array. However, the proposed method was a general method, so it can be applied to recognize other scripts. Soumya and Kumar [11] focused on recognizing the ancient script in the time of King Ashoka (Brahmi script) using Gabor features and zone-based feature extraction methods to extract the features of these characters and ANN for the classification of the extracted features. The accuracy of this system was 80.2%. Next, Vellingiriraj, et al. [6] focused on developing a Brahmi word recognition system in which they used zonal density to extract the features and ANN to classify the extracted features. Three types of samples: vowels, consonants, and consonantal vowels (compound characters), were used to measure the performance of the system, achieving accuracies of 93.45%, 93.45%, and 90.24%, respectively. Apart from Brahmi word recognition, in 2016, Gautam, et al. [5] focused on Brahmi character recognition. The zone method was used to extract the features of the Brahmi characters, and the template matching method (coefficient correlation) was used for the classification of all extracted features, achieving an accuracy of 88.83%. However, this method could not recognize non-connected characters. Another work also developed a system to recognize Brahmi characters [4], focusing on the geometric elements in each zone (corner point, bifurcation point, ending point) and the geometric features (corner point, bifurcation point, ending point, intersect point, circle, and semi-circle) to extract the features. In addition, some sets of rules were created for the classification of all these features, yielding a final accuracy of 94.10%. Existing Brahmi word recognition systems have performed well; however, their performance can also be increased. In the recently, Gautam, et al. [12] used DCNN to recognize the Brahmi words and achieved 92.47% accuracy, which is very good.

Apart from that, recently, few works also can see on Brahmi script such as Brahmi script to Pali language translation [10], Brahmi word segmentation [13]. All these studies completed on local Brahmi dataset. Thus, Gautam, et al. [14] published standard dataset for Brahmi character and word recognition. Therefore, the literature review also focused on finding a methodology to further improve the accuracy of existing Brahmi word recognition systems.

Feature extraction and classification are the most important steps of OCR, which have shown excellent performance in recognising various scripts such as Bangla script, Devanagari script, Kannada script, Telugu script, Tamil script, Malayalam script, Gurumukhi script, Gujarati script, and Sinhala script. The literature reviewed in this study focused on the feature extraction methods and classifiers based on the performance of the character, words, and text recognition of all scripts under the Abugida world writing system to which these scripts and the Brahmi script belong.

Some studies also focused on Bangla compound character and word recognition. For example, zone, and row-wise longest-run with SVM (RBF) [15] were used for the feature extraction of Bangla compound characters, achieving 72.87% accuracy. Structural decomposition was also used to recognise Bangla compound characters, gaining 96.17% accuracy, while template matching was used to classify the features [16]. When compound characters are combined in a meaningful manner, a word is created. a study used HMM to recognise Bangla words and reached 85.49% efficiency [17]. Bhowmik, et al. [18] applied the elliptical feature extraction method and various classifiers like the Naïve Bayes, Bagging, Dagging, SVM, MLP, to evaluate the performance of the classifiers, achieving accuracies of 74.41%, 60.00%, 69.41%, 77.35%, and 77.94%, respectively. Therefore, it can be concluded that SVM and ANN performed better in recognising Bangla words compared to the other classifiers. HOG for feature extraction and ANN for the classification of the extracted features were used to recognise Bangla words in one study, which achieved an accuracy of 87.35% [19].

Some studies focused on the comparison of feature extraction techniques and classifiers to recognise Devanagari characters. For example, In contrast, the accuracy of ANN was found to be 97.15%, and SVM returned a 95.64% accuracy with the Histogram Oriented Gradient (HOG) used for extracting the features of Marathi characters (Devanagari characters) [20]. ANN and SVM were also used to classify the characters of Hindi characters (Devanagari characters), where ANN achieved 92% accuracy and SVM obtained 90% accuracy [21], so ANN performed better than SVM in the study. SVM, ensemble subspace discriminant, k-NN, Weighted k-NN, and bagged tree classifiers were used to classify the features of Hindi characters (Devanagari characters), with HOG as the classifier and the projection profile, achieving an accuracy of 96.6%, 94.7%, 93%, 93.1%, and 93.2%, respectively, indicating that SVM performed better than the other classifiers [22]. Chandure and Inamdar [23] focused on the comparison of KNN, ANN, and SVM and obtained an accuracy of 87.5%, 70%, and 87.5%, respectively in classifying the features of Devanagari characters extracted using the chain code method [23]. Therefore, it can be seen that SVM performed well compared to ANN and reached the highest accuracy in recognising the Devanagari characters. Similarly, Ansari and Sutar [24] compared the performance of SVM, KNN, and ANN to recognise Devanagari characters, which were extracted using geometric features, regional features, gradient features, and distance transforms features. The ANN obtained an accuracy of 91.3% whereas SVM and KNN obtained 86.34% and 79.1% accuracy, respectively. Hence, ANN had a higher recognition rate compared to SVM and KNN.

In recent times, a few studies have focused on recognising the Telugu script. For example, binarization and zone with the nearest neighbourhood [25], glyph, the zone with a hierarchical classification system [26] and Mojette transform, together with PCA with ANN [27] were used to recognise Telugu characters, achieving 78%, 88.15%, and 98.10% accuracy, respectively. Sastry, et al. [25] applied the zoning method and 2D FFT feature extraction method to assess the performance of a Telugu character recognition system with the nearest neighbour classification, obtaining 78% and 65% accuracy, respectively. Hence, the zoning method performed better than the 2D FFT method.

Based on the literature on Telugu script recognition, PCA performed best in extracting the features while ANN proved to be the best classifier.

Discrete cosine transform and ANN [28], HOG and SVM [29], and binary image, zoning method, and template matching [30] were used for the recognition of Kannada numerals, achieving an accuracy of 90%, 95%, and 98%, respectively. The feature of the Kannada characters was extracted via the structural elements as horizontal line, vertical line, left curve, right curve, upward curve, and the downward curve in each zone [31], chain code with SVM [32], achieving an accuracy of 85.43%, 87.24% respectively.

Structural features (speeded up robust features, curvature feature, diagonal feature) with ANN and SVM together [33], HMM [34], structural characteristics (zoning method, diagonal zone, horizontal zone, vertical zone) with ANN [35], and structural features (corner, edge, Boundary) with SVM [36] were used to recognise Malayalam characters, achieving 81.10%, 81.38%, 93.20%, and 95.60% recognition rates, respectively.

Therefore, structural elements were the best feature extraction method for Malayalam character recognition. Further to this, Varghese, et al. [37] suggested a theory-based method based on structural elements (corner, ending, bifurcation, loop features) and introduced some sets of rules to recognise Malayalam characters. However, the study imposed 19 conditions before this method could be applied for Malayalam character recognition. Similarly, Raveena, et al. [38] also suggested a theory related to structural elements (length of character in horizontal and vertical, number of endpoints, number of intersections in horizontal & vertical, number of loops, direction) for feature extraction and SVM for the classification of Malayalam characters. Chacko and Dhanya [39] applied zoning density, projection profile, chain code features, and HOG for feature extraction and compared the performance of all feature extraction methods, with ANN as the classifier. The accuracy achieved using zoning density, projection profile, chain code features, and HOG was 84.6%, 88.07%, 78.8%, and 94.23%, respectively, indicating that HOG recorded the highest accuracy. Similarly, zoning density, background directional distribution, profiles, HOG features with SVM was used for recognition of Gurumukhi characters and achieved 97.25%, accuracy [40].

In 2017, zoning features, raster diagonal features, horizontal peak extent features, and the combination of all three feature extraction methods (zoning features, diagonal features, horizontal peak extent features) were applied to assess the performance of Gurumukhi character recognition, achieving an accuracy of 92.08%, 91.63%, 81.34%, and 87.34%, respectively, indicating that the zoning method was the best feature extraction method. A study compared the accuracy of chain code and geometric method for the feature extraction of Gujarati characters and obtained 87.29% and 96.65% accuracy, respectively, with SVM used for the classification of the extracted features [41]. Mahto, et al. [42] also compared the performance of SVM and KNN and found that SVM performed better than KNN with an accuracy of 98.06% versus 92.80%.

The density of each zone [43] geometric features (connected and disconnected components, endpoint, closed loop) [44], and other geometric features (endpoint, joints, lines, left curves, right curves, circles) [45] as feature extraction methods were applied to extract the features of Gujarati characters, achieving accuracies of 86.66%, 88.78%, and 95%, respectively, showing that geometric features were the best feature extraction method to achieve a high recognition rate.

Sharma, et al. [46] compared some classifiers such as Naive Bayes and ANN in classifying the features of Gujarati numerals, extracted via the zoning method. The study obtained 85.60% and 95.92% accuracy for Naive Bayes and ANN, respectively, indicating that ANN is better than KNN in recognising the Gujarati numerals [46]. Apart from the recognition of numbers and characters in the Gujarati script, Solanki and Bhatt [47] applied PCA and ANN together to achieve a 93.25% accuracy in recognising Gujarati words.

Only a few studies have attempted to build a recognition system for Sinhala script because the structure of Sinhala characters is quite complex and is a tough task to recognise [48]. In recent times, Silva, et al. [48] used contour tracing to extract the features of Sinhala characters and achieved 53% accuracy. The zoning method [49], geometry, zoning, and structural elements [50] were used to extract the features of Sinhala characters, while ANN was used for the classification of the extracted features, achieving 71.7%, and 82.1% accuracy, respectively. Therefore, it can be surmised that the combination of geometry, zoning, and structural features was the best method for extracting features of Sinhalese script.

Similar to the Sinhala script recognition, only a few studies were completed in the area of Odia character recognition. The HOG method performed well in recognising various scripts, and similarly, the HOG method was used with SVM to recognise Odia characters, reaching 97.2% accuracy [51]. Distance-based features, centroid features, shadow features with Euclidean distance [52], and local binary pattern with KNN [53] were used to recognise Odia characters, achieving 87.6%, and 96.5% accuracy, respectively. Therefore, HOG and SVM were the best feature extraction method and classifier for recognising Odia characters.

In conclusion, various feature extraction methods such as PCA, HOG, geometric features, zoning method, zonal density, chain code, projection profile, Gabor features were found to perform well in recognising various scripts whereas SVM and ANN achieved the highest accuracy. Hence, this study focused on checking the performance of various feature extraction methods and evaluating the performance of the ANN and SVM classifiers in recognition of the features of Brahmi words

4. Dataset

Standard Brahmi dataset is available for Brahmi characters, Brahmi words recognition [14]. Brahmi dataset is the combination of the various types of samples of characters. In the dataset, 170 classes of a total number of characters where 27 varieties of consonants, 4 types of vowels, and rest (139) belongs to compound characters. As the knowledge, Brahmi script has 368 types of characters [4] but could not find all kind of characters because of the lack of resources. The database consists of printed Brahmi words composed of 6,475 images of 170 classes for the training and 536isolated characters of 170 classes for the testing. This dataset also has 55 samples of Brahmi text (which are not isolated) to check the performance of segmentation methods in the future.

5. Methodology

The recognition of a character image undergoes segmentation (line and character detection), pre-processing (grayscale, binarization, and size normalisation), feature extraction, and classification under the OCR system. However, all characters have already been isolated from the Brahmi word dataset, so the segmentation method is not required in this case. Feature extraction and classification are the decision-making steps in supervised learning. The steps of Brahmi word recognition under supervised learning are input, pre-processing, feature extraction, classification, and output. The Brahmi dataset was used to evaluate the performance of the Brahmi word recognition system.

5.1. Input image

Input is the first phase of image processing or pattern recognition tasks. In this study, the image of Brahmi words in JPG format was obtained and treated as the dataset in the first step. The size and resolution of the pictures were not fixed, so a pre-processing step was applied to enhance the quality of the samples and to convert it into useful content.

5.2. Pre-processing

Pre-processing is an essential step for enhancing the visibility and readability of input. The pre-processing steps employed by this study are thresholding and size normalisation (resizing), further elaborated below.

4.2.1 Binarization: Grayscale image can be converted into binary image by Binarization (thresholding). The main motive of thresholding techniques is to enhance the visibility of edge, in which case, the shape of the region is more important compare to the intensities of pixels. Two approaches: global threshold, local threshold can use under binarization. A single threshold value is used for the whole image according to the background of the image in the global threshold. Hence, this method is useful if background of the image is simples and uniform. Although, various values threshold for each pixel is chosen based on the data of local area in local threshold. Usually, this approach is used for real-life documents because, sometimes, these documents are designed deliberately with stylistic, colourful, and complex backgrounds.
The background of all images in the Brahmi dataset was white and therefore not complicated. Hence, a Global threshold approach was used to perform the thresholding process in this study.

4.2.2 Size normalization (resizing): Normalisation was applied to obtain all characters of uniform size. The characters varied in size, so their size should be made uniform to enable comparison. For example, the input of ANN and SVM should be in vector form of a fixed size. Normalisation should reduce or increase the original size of the image without changing the structure of the image.

The performance of each feature extraction can change according to the size of the samples. Therefore, the size of the chain code, the Gabor filter, the geometric feature, the HOG, the PCA, the projection profile, the zonal geometric feature extraction, and the zonal density feature extraction in this study was set to 32 × 32, 40 × 48, 100 × 80, 32 × 32, 32 × 32, 100 × 80, 100 × 80, and 36 × 27 pixels, respectively

5.3. Feature extraction

A useful feature of the character will increase the accuracy of character recognition. Humans also use features to identify a character. The features that humans use to identify characters can be similar or dissimilar to the features that machines use to identify characters. For example, shape -based features of a character can be an identifier of the character for both humans and machine. However, the transform-based approach helps only machines to identify a character because this approach converts images into signals to complete the recognition process. This signal cannot be seen by humans, so this approach is only useful for machines. Few characters can have similar structure features so, classifier can be confused to differentiate these types of characters. Thus, transform-based features can be more useful to identify those characters whose are similar in shape.

More details of selected feature extraction techniques by literature review are discussed below.

4.3.1 Chain code: The chain code method has shown effective performance in representing the shape of different objects. The two most popular types of chain coding are 4- and 8- directional chain codes. The value of the chain code defines 0-3, or 0-7 according to the grid whether it is 4-connected or the 8-connected.

This study focuses on the Freeman chain [54] under the 8-directional chain code method.

The chain codes take a path of the outer edge of a character in a clockwise direction in which each corner is allocated a shape with the direction code (0-7) indicating the direction of the next corner. The same processes are applied until the last point of the path no longer touches a starting point of the way.

4.3.2 Gabor filter: The Gabor filter (GF) can be used to model cells in the visual cortex of mammals and is capable of detecting strokes of particular orientation even those of short length. Spatial and spatial-frequency domains are used to achieve the joint optimal resolution of GF.
The feature extraction algorithm is outlined below:

1. I_x (m,n), the given input image, is read
2. The Gabor filter bank is applied on I_x (m,n)
3. Output of the previous step and Even Symmetric Filter are combined.
4. By using the output of the convolution of I_x (m,n), standard deviation is computed

In conclusion, 30 (5 frequency × 6 orientations) values are used to create the feature vector which are further used to recognize the characters.

4.3.3 Geometric feature: The geometric method is a high-level feature, which is partially inspired by the way humans process patterns. Instead of merely describing an image using individual pixels or their statistical groupings, the general shape of an object is used as a feature. For example, the number of loops, junctions, crossings, and the endpoint is used to classify patterns [4].

This study focused on six geometric features: the number of endpoints, the number of bifurcation points, the number of corner points, the number of intersecting points, the number of circles, and the number of semi-circles. This is because these geometric features have been proven to perform well in recognising Brahmi characters [4].

The number of geometric features stored in a vector is called a feature vector of the character, which is shown by f_n (Equation 5.6). This procedure is sequentially repeated for all characters.

$$F_n = [E, B, C, I, CR, SCR] \quad (1)$$

Where f denotes the feature vector of the character and n is the number of characters. Meanwhile, $E, B, C, I, CR,$ and SCR denote the number of endpoints, the bifurcation points, the corner points, the intersecting points, the circles, and the semi-circles, respectively. The combination of all values of the features yields a feature vector (f_n). All feature vectors of all characters were combined to create a set, which is used for classification.

4.3.4 Histograms of Oriented Gradients (HOG): The Histograms of Oriented Gradients (HOG) is a feature descriptor that is computed by counting the occurrences of gradient orientations in localised parts of an image. To calculate these features, the image was divided into cells where the histograms of gradient directions from each of these cells.

To compute the HOG, first, the image was resized to 32 × 32 pixels. Then, the character was divided into 8 × 8 blocks. These blocks are called cells. A square sliding window of 2 × 2 was considered with a 50% overlap. A cell was taken and the magnitude at each point of that cell calculated. According to HOG, nine ‘bins’ (10, 30, 50, 70, 90, 110, 130, 150, and 170) have to be defined. Therefore, the calculation for feature values will be (total number of blocks) × (4 × number of bins). Here, the total number of blocks was 7 × 7 because of the 2 x 2 sliding windows and the 50% overlap.

The feature values by the HOG of a character 7 × 7 × 4 × 9 = 1,764. These values are put in a sequence called the feature vector. The same processes were applied to extract the HOG features of all characters to create a set. This set was then used to classify the characters.

4.3.5 Projection profile: The projection profile is an active and popular method for text pre-processing used for a word or character segmentation as well as the determination of the orientation of the scanned text.

Projection profile depend on the black pixels of each rows and column of a character and it is valuable to recognize the various characters [22].

This study focused on the vertical and horizontal profile by including the pixels that were vertical and horizontal, which, together, formed a one-eighty dimension feature vector (f_n), with 80 pixels belonging to the horizontal pattern and 100 pixels belonging to the vertical profile. Equation 5.8 shows the calculation of the dimension feature vector:

$$f_n = [P_h, P_v] \quad (2)$$

Where f denotes the feature vector of the character and n is the number of characters while P_h and P_v denote the horizontal profile and vertical profile of the character, respectively (Equation 2). The combination of all values of features will result in a feature vector (f_n). All feature vectors of all characters were combined to create a set that was later used for classification.

4.3.5 Principal component analysis (PCA): PCA is based on the dimensional reduction technique which focus to low dimensional space from the high dimensional space. The steps to compute the PCA in this study included taking a character, calculate the pixel value of character and mean value of the character, subtracting both values, and storing the values in a newly matrix. Later on, the covariance matrix of the new matrix was computed, and the eigenvectors and eigenvalues of that covariance matrix calculated. Once the eigenvectors were derived from the covariance matrix, the next step was to adjust all eigenvalues values into maximum to minimum to become a vector called the feature vector. If there are n dimensions of the data, the feature vector will take the form shown by Equation 3:

$$\text{Feature vector} = [\text{eig}_1, \text{eig}_2, \dots \dots \dots \text{eig}_n] \quad (3)$$

Where, eig_n is the eigenvalue of the nth row. All feature vectors of all characters were combined to create a set that was then used for classification.

4.3.7 Zonal geometric feature extraction method: The zonal density feature extraction method is a hybrid method combining zone feature extraction and geometric feature extraction. In this method, first of all, each character is divided into zones, and then the geometric features will extract the features from each zone. This study focuses on three geometric features: the number of endpoints, the number of bifurcation points, and the number of corner points because these features have performed well in recognising Brahmi characters in the past [4].

The character size in this method was 100 × 80, which was divided into 5 × 5 zones, and the size of each zone was 20 × 16. Hence, 25 types of features were used to recognise the Brahmi words in this study.

The number of geometric features of each zone stored in a vector is called a feature vector of the character, as shown by f_n (Equation 4). This procedure is sequentially repeated for all characters.

$$f_n = [Ez_1, Bz_1, Cz_1 \dots \dots \dots Ez_{25}, Bz_{25}, Cz_{25}] \quad (4)$$

Where f denotes the feature vector of the character and n is the number of characters. Meanwhile, Ez_1 denotes the end point, Bz_1 shows the bifurcation point, and Cz_1 represents the corner point of the first zone. Similarly, a total of 75 features were calculated from the twenty-five different zones of a character. The combination of all values of features yielded the feature vector (f_n). All feature vectors of all characters were combined to create a set, where the set was later used for classification.

4.3.8 Zonal density feature extraction method: The zonal density technique is a combination of zoning method and density feature extraction. According to this approach, firstly, the image of a character divide into several zones, furthermore, zonal density (X) follow to calculate the density of each zone.

$$X = \frac{\text{Number of foreground pixels of each zone}}{\text{Total number of pixels in each zone}} \quad (5)$$

The character size for this method was 36 × 27 pixels, divided into 9 × 9 zones, with the size of each zone being 4 × 3. So, 81 types of features were used to recognise the Brahmi words in this study.

The density value of each zone stored in a vector is called a feature vector of the character, as shown by f_n (Equation 6). This procedure was sequentially repeated for all characters.

$$f_n = [dz_1, dz_2, dz_3 \dots \dots \dots dz_{80}, dz_{81}] \quad (6)$$

Where f denotes the feature vector of the character and n is the number of characters. Meanwhile, dz_1 denotes the density of the first zone. Similarly, a total of 81 features were calculated from the 81 different zones. The combination of all values of features yielded a feature vector (f_n). All feature vectors of all characters were combined to create a set, which was later used for classification.

5.4. Classification

A classifier takes the extracted features, which obtained by various methods, and use to take the final decision of the whole recognition system. The classifier helps to recognize the input characters and the output of the class to which the characters belong.

According to the literature of this study, ANN and SVM has performed well to identify the various scripts. So, this study was used ANN and SVM to classify all features.

4.4.1 Artificial neural network (ANN): ANN model is inspired by biological neural networks and several types of theories are linked to the working process of biological neural networks, but the main characteristics were determined and implemented for artificial networks.

In this study, a feed-forward back-propagation neural network with two hidden layers was took to run the classifier. The hidden layers used a log-sigmoid activation function, and the output layer was based on a competitive layer, as one of the characters had to be recognized. The number of input neurons was determined by the length of the feature vector X_m. The neural network was then chosen and presented. The parameters of the neural network are mentioned in Table 1:

For this experiment, 170 classes of characters and 25 samples of each class were developed. Therefore, complete 5,025 responses were considered for training the system. After that, the features were extracted from all sample characters called feature vectors. The size of a feature vector is 1 × N (different feature extraction methods have different dimensions), so the size of the whole input sample was 5025 × N for the training phase. In creating a neural network, a target vector is required to provide input data as well as output data in the form of a vector. The size of the target vector was 1 × 5025 in this study. The values of the target vector started from one (1) to 170.

Table 1: Network training parameters

Parameter	Value
Input nodes	N
Number of hidden layers	2
Hidden nodes	85, 85
Training algorithm	LM
Perform function	MSE
Training goal achieved	0.01
Training epochs	100
Validation checks	10

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2020 doi:10.20944/preprints202006.0048.v1

Learning rate	0.1

4.4.2 Support vector machine (SVM): SVM is a statistical method that has shown extraordinary performance in character recognition [15]. A set of input data and target set (predict) use for each provided input by standard SVM. Basic SVM is useful for the isolate the linearly dataset and it also be able to apply on non-linear datasets by using by indirectly mapping. The nonlinear inputs into a linear feature space where the maximum margin decision function is approximated. Kernel function is valuable to complete the mapping process. SVM has various types of kernels that can overcome the problem of non-linear data; these kernels have shown excellent performance as well so, various types of kernels are also used to evaluate the performance and reach the highest performance of the SVM classifiers. Apart from the kernels, training set and target vectors are also compulsory when using SVM for the classification.

To create a training set, the size of a feature vector was set to N × 1 (different feature extraction methods have different dimensions). Therefore, the size of the whole input sample was N × 5025 for the training set. For classifying by the SVM, a target vector is required to provide input data as well as output data in the form of a vector. The size of the target vector was set to 5025 × 1 in this study. The values of the target vector started from 1 to 170.

6. Result

6.1. Performance evaluation

The description of the performance is divided into two parts. The achieved accuracy of ANN is discussed followed by the performance of SVM with the feature extraction methods chain code, Gabor filter, geometric method, histograms of oriented gradients (HOG), Principal Component Analysis (PCA), projection profile, zonal geometric feature extraction method, and zonal density feature extraction).

Accuracy of this algorithm was estimated in accordance with the given expression: If out of Y characters, X is recognized correctly, then accuracy of the algorithm is: X/Y × 100%.

Using ANN as the classifier for Brahmi words, the accuracy of chain code, Gabor filter, geometric features, PCA, projection profile, zonal geometric elements, HOG and zonal density features was 46.27%, 23.76%, 26.18%, 12.26%, 20.59%, 45.76%, 77.26%, and 68.69%, respectively.

Using SVM as the classifier for Brahmi words, the accuracy of chain code, Gabor filter, geometric features, PCA, projection profile, zonal geometric elements, HOG and zonal density features was 73.84%, 77.41%, 46.12%, 32.34%, 83.29%, 81.88%, 94.35% and 92.82%, respectively.

Therefore, Gabor filter, geometric features, PCA, projection profile, and zonal geometric elements did not perform well with ANN as the classifier for all extracted features. Compared to these feature extraction methods, the performance of HOG and zonal density features was better and achieved better accuracies. Similarly, the accuracies of Gabor filter, geometric features, PCA, projection profile, and zonal geometric features did not yield good results with SVM as the classifier for all extracted features. However, the accuracy was increased when HOG and zonal density were used to extract the features of the Brahmi words, and SVM was used to classify the extracted features. Hence, HOG and zonal density features were the best feature extraction methods. Similarly, SVM was a better classifier than ANN.

Values of the HOG, zones of each character and size of the zone are important factor for Brahmi word recognition to increase the accuracy because values of the HOG can help to achieve highest performance of the HOG method, and zones of each character and size of the zone is valuable to reach maximum performance of zoning method.

Value of the number of blocks can be changed 4 × 4, 8 × 8, 12 × 12, 16 × 16, 20 × 20. The feature values by the HOG of a character 3× 3 × 4 × 9 = 324. These values are put in a sequence called the feature vector. Achieved accuracy of this parameter was 64.39% and 86.24% to used ANN, SVM respectively for the classification. The feature values by the HOG of a character 11× 11 × 4 × 9 = 4,356. These values are put in a sequence called the feature vector. Achieved accuracy of this parameter was 69.71% and 88.61% to used ANN, SVM respectively for the classification. The feature values by the HOG of a character 15× 15 × 4 × 9 = 8,100. These values are put in a sequence called the feature vector. Achieved accuracy of this parameter was 66.39% and 87.83% to used ANN, SVM respectively for the classification. The feature values by the HOG of a character 19× 19 × 4 × 9 = 12,996. These values are put in a sequence called the feature vector. Achieved accuracy of this parameter was 51.39% and 68.45% to used ANN, SVM respectively for the classification. As can see that the performance of the system is decreased compare to previous performance of HOG so, the analysis of HOG stopped here.

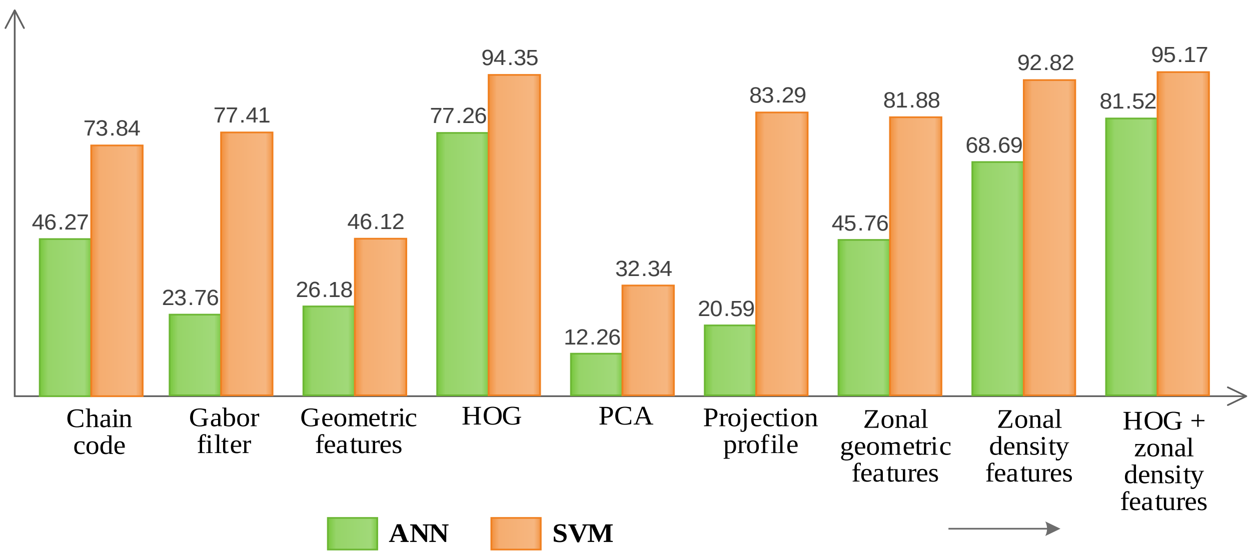

Figure 2: Performance of techniques (supervised learning) in recognizing the Brahmi words.

Comparison of all block size of HOG compared and find out that 8× 8 block, where size of features 1,764 with SVM achieved highest accuracy compare to other combination. All details of upper discussed is presented in table 2.

Table 2: Performance of the HOG technique in recognizing the Brahmi words

Block size	Size of Feature Vector	Accuracy	
		ANN	**SVM**
4 × 4	324	64.39%	86.24%
8 × 8	1,764	77.26%	94.35%
12 × 12	4,356	69.71%	88.61%
16 × 16	8,100	66.39%	87.83%
20 × 20	12,996	51.39%	68.45%

Number of zones of each character and size of the zone is very important entities which can affect the performance of zoning density method. Means, it can say that if the number of zones of each characters and size of the zone will change so, accuracy of the system is can be changed so, this study is focused on the analysis of these two entities. Number of zones of each character can be changed 2x2, 3x2, 3x3, 4x3, 5x3, 4x4, 6x4, 5x5 and size of the zone 3x3, 4x4, 5x5, 6x6, 7x7, 8x8, 9x9, 10x10, 11x11 where ANN and SVM used for classification. All achieved results by using described zoning size and classifiers are presented in table 3.

Whole analysis of zoning density method it can see that system achieved 68.69%, 92.82% accuracy by ANN and SVM, respectively, for the classification while image size is 36x27 where number of zones of each character and size of the zone is 4x3, 9x9.

Parameter of the ANN plays an important role to reach maximum performance of any network and the value of learning rate, number of hidden layers can affect the performance of the recognition system. 0.9, 0.1, 0.01, and 0.001 value used as learning rate for the analysis of the performance of ANN for Brahmi words recognition. Brahmi word recognition system achieved 81.81%, 86.34%, 92.82%, and 77.92% respectively. It can see that starting accuracy of the recognition was 81.81% while learning rate was 0.9. Performance of the system was increased by 4.53% and become 86.34% while value of learning rate was 0.1. Again, the accuracy of the recognition system further increased by 6.48% while 0.01 value used as learning rate. Although, the recognition rate decreased by 14.90% while value of learning rate was 0.001. As can see that the performance of the system is increased compare to previous performance so, the analysis of learning rate stopped here. So, in the conclusion, it can say that Brahmi word recognition system achieved highest accuracy when 0.01 was the learning rate.

1, 2, 3, and 4 value used as hidden layer for the analysis of the performance of ANN for Brahmi words recognition. Brahmi word recognition system achieved 83.44%, 92.82%, 85.38%, and 78.12% respectively. It can see, starting accuracy of the recognition was 83.44% while hidden layer was 1. Performance of the system was increased by 4.53% and become 86.34% while value of learning rate was 2. Again, the accuracy of the recognition system further increased by 6.48% while 3 value used as learning rate. Although, the recognition rate decreased by 14.90% while value of learning rate was 4.

As can see that the performance of the system further increased compare to previous performances of hidden layers so, the analysis of hidden layer stopped here. In the conclusion, it can say that Brahmi word recognition system achieved highest accuracy when 0.01 was the learning rate.

As already mentioned, basic SVM is useful for the classification of linear data. However, it can be possible, the feature vector cannot be linearly separable, and this issue can be minimizing by kernel. Kernel function is able to be mapped into high-dimensional feature space from original input space. Furthermore, a suitable kernel function is important to reach the highest accuracy of SVM classifier. In previous studies, various types of kernel function (polynomial, radial basis function, Mahalanobis, and sigmoid) have been applied to enhance the performance of the recognition system. So, this study applies these types of kernel to build the Brahmi word recognition system and compare the performance

Table 3: Achieved accuracy by zonal density method by several separation of the image in zones using

		Number of zones in each character (achieved accuracy by ANN)								
		3x3	**4x4**	**5x5**	**6x6**	**7x7**	**8x8**	**9x9**	**10x10**	**11x11**
Size of each zone in a character	**2x2**	30.68%	31.74%	49.12%	39.41%	53.13%	54.24%	43.71%	64.68%	51.39%
	3x2	39.41%	32.84%	44.83%	46.25%	51.83%	62.39%	61.24%	62.59%	57.26%
	3x3	36.89%	46.26%	38.49%	47.91%	49.67%	61.26%	62.48%	58.24%	52.68%
	4x3	28.12%	44.51%	44.86%	53.64%	58.31%	53.64%	68.69%	59.61%	49.81%
	5x3	34.26%	52.17%	51.74%	48.32%	62.41%	64.58%	62.39%	51.38%	52.14%
	4x4	53.14%	54.68%	59.45%	51.68%	51.02%	49.81%	59.42%	41.26%	41.23%
	6x4	49.86%	49.87%	56.37%	46.14%	49.34%	41.69%	43.21%	49.64%	46.24%
	5x5	37.64%	35.31%	42.91%	41.68%	34.18%	38.61%	52.62%	33.64%	34.47%

		Number of zones in each character (achieved accuracy by SVM)								
		3x3	**4x4**	**5x5**	**6x6**	**7x7**	**8x8**	**9x9**	**10x10**	**11x11**
Size of each zone in a character	**2x2**	41.37%	38.42%	36.49%	48.16%	53.67%	73.43%	79.79%	81.45%	78.29%
	3x2	38.62%	40.33%	61.21%	81.37%	86.61%	88.19%	88.94%	79.12%	76.36%
	3x3	38.41%	48.62%	59.54%	78.24%	87.29%	83.42%	87.02%	75.47%	76.21%
	4x3	43.87%	51.41%	59.64%	81.61%	84.38%	81.47%	92.82%	73.26%	71.83%
	5x3	48.63%	53.62%	57.14%	71.68%	65.24%	51.36%	56.84%	51.62%	53.85%
	4x4	52.16%	58.39%	68.76%	74.34%	71.21%	68.39%	43.26%	51.20%	36.84%
	6x4	46.76%	63.48%	58.81%	52.19%	32.58%	41.26%	43.25%	38.51%	34.66%
	5x5	41.29%	53.71%	48.67%	50.22%	51.26%	46.38%	41.26%	49.37%	35.21%

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 5 June 2020 doi:10.20944/preprints202006.0048.v1

- Polynomial kernel with degree d is mentioned in eq 7:

$$K(x_i, x_j) = (yx_i^T + 1)^d, \; y > 0 \quad (7)$$

- Radial basis function kernel is mentioned in eq 8:

$$K(x_i, x_j) = exp(-y\|x_i - x_j\|^2), \; y > 0 \quad (8)$$

- Sigmoid kernel function is mentioned in eq 9:

$$K(x_i, x_j) = tanh(yx_i^T x_j + r) \quad (9)$$

Multi-Layer Perception Kernel or Hyperbolic Tangent Kernel is another name of Sigmoid kernel function where, the value of the g, r and d can be adjusted according the data.

- The Mahalanobis kernel is mentioned in eq 10:

$$K(x_i, x_j) = -\frac{\delta}{m}(x_i - x_j)^T Q^{-1}(x_i - x_j) \quad (10)$$

Mahalanobis distance can be controlled by the scaling factor δ (> 0), and Q, m represents the covariance matrix, and dimension of the Brahmi dataset, respectively. Upper discussed formula is used to calculate the accuracy and achieved accuracy are presented in table 4.

Table 4: Classification results with SVM using different kernel functions

Kernel Type	Features	Accuracy
RBF	HOG + zoning density features	95.17%
Poly		93.20%
Sigmoid		93.89%
Mahalanobis		94.48%

As can discussed earlier, HOG and zonal density feature extraction method and ANN, SVM performed (accuracy) well to recognize the Brahmi words. Earlier, parameter of ANN and SVM checked and achieved results is notify that the combination of 0.01 learning rate and 2 number of hidden layers achieved highest accuracy compare to other values of parameter. Similarly, RBF kernel type achieved highest accuracy.

HOG and zonal density feature extraction methods had the best performance. Thus, both feature extraction methods were combined to achieve an accuracy of 81.52% and 95.17% with ANN and SVM as classifiers, respectively while 0.01 learning rate and 2 number of hidden layers parameter used for ANN and RBF kernel for SVM.

Few studies completed in the Brahmi characters and word recognition. Such as, template matching [5], geometric features with set of rules [4], Gabor filter and zoning methods with ANN [11], and zonal density features with ANN [6]. All these studies used local dataset and dataset was not common in all studies. So, for the comparison of previous studies with proposed study, this study applied previous used techniques on new dataset and compare the performance of proposed study, which are presented in table 5.

Table 5: Performance of the previous applied supervised techniques in recognizing the Brahmi words (a common dataset used)

Method	Accuracy
Template matching	46.73%
Geometric features with set of rules	26.18%
Gabor filter and zoning methods with ANN	39.84%
Zonal density features with ANN	68.69%

As can see in the figure 2 and table 5, the zonal density approach performed well compare to other previous used approaches and achieved 68.69% accuracy. Furthermore, HOG+ zonal density with SVM achieved 92.47% accuracy, which is better than all previous studies (a common dataset used).

6.2. Comparison with other studies

The performance of HOG, zonal density features, and a combination of HOG + zonal density features along with SVM were compared to previous studies on Brahmi word recognition such as zonal density with ANN [6], and Gabor filter + zonal structural features with ANN [11]. The achieved accuracy was 94.35%, 92.82%, and 95.17% for HOG, zonal density features, and the combination of HOG + zonal density features along with SVM, respectively. Moreover, zonal density with ANN [6], and Gabor filter+ zonal structural features [11] achieved 80.20% and 91.57% (average), respectively. Hence, it can be concluded that the HOG + zonal density feature extraction methods along with SVM were better than the techniques used in previous studies.

Vellingiriraj, et al. [6] used zonal density with ANN and achieved 91.57% (average) accuracy. Similarly, zonal density with SVM was used in this current study, achieving 92.82% accuracy. It can be seen that the feature extraction method is common in both studies, albeit with a different classifier. Therefore, it can be said that SVM performed better compared to ANN in recognising Brahmi words. The performance of various approaches related to Brahmi word recognition, along with the proposed method, is outlined in Figure 3. It can be observed that HOG + zonal density features, along with SVM, achieved the highest accuracy of all other studies.

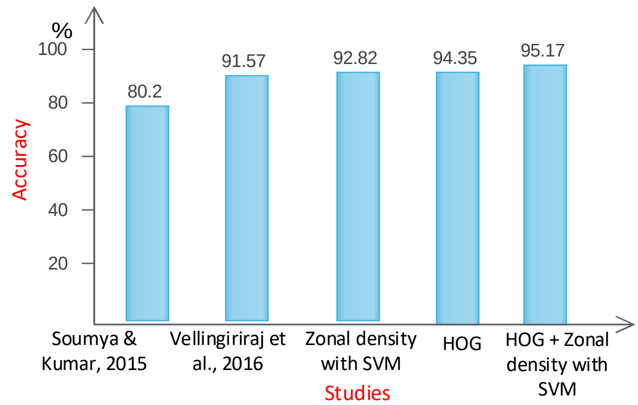

Figure 3: Comparison of performance of supervised learning with previous studies in recognizing Brahmi words

7. Conclusion

This is the first application where various feature extraction methods and classifiers are used to recognized to Brahmi words and according to analysis, two feature extraction methods combine in the domain of Brahmi words recognition. Combination is performed at the feature level as well as decision level by using classifiers. Encouraging results are obtained from the experiments. High accuracies in the range of 95.17% have been achieved by using combination techniques as shown in the previous Result section. There is an increase of over 3.6% with the best performing HOG+ zonal density with SVM classifier when RBF kernel is used as the classifier for 536 samples of 170 different classes. So, this model proves to be useful for this complex pattern recognition problem and makes a better decision based on the information provided by the base classifier.

Though, in the present work, two sources of information with different feature sets have been combined using their respective classifier results but this process can be extended to include more input sources along with different feature extraction techniques and classifier. With the increase in the number of sources, an intelligent and dynamic selection procedure needs to be employed in order to facilitate combination in a more meaningful way. The combination being an overhead to the classification task, it is important to develop methods that can indicate if the combination would work or not qualitatively. In future, the work can be extended for a larger dataset so that the robustness of the procedures can be established. The Brahmi word recognition system here is a general framework which can be applied to other similar pattern recognition tasks to establish its usefulness in document analysis research. Furthermore, the proposed system was quite efficient (in terms of accuracy) and effective at performing the recognition and classification. Furthermore, future work of this study to develop a handwritten Brahmi word dataset and handwritten Brahmi words recognition system.

References

[1] A. N Roy, J. Jose, A. Sunil, N. Gautam, D. Nathalia, and A. Suresh, "Prediction and Spread Visualization of Covid-19 Pandemic using Machine Learning," *Preprints* 2020, 2020050147 (doi: 10.20944/preprints202005.0147.v1), 2020.

[2] R. K. Budhi, D. T. Hidayat, and J. P. Pangapul, "Development of English-Indonesia Idiom Mobile Dictionary Using Optical Character Recognition," *Jurnal Transformatika*, vol. 16, no. 1, pp. 14-21, 2018.

[3] N. Gautam, R. Sharma, and G. Hazrati, "Akkhara-Muni: An instance for classifying PALI characters," in *International Conference on Computational Intelligence and Communication Networks*, Jabalpur, India, 2015: IEEE, pp. 252-253.

[4] N. Gautam and S. S. Chai, "Optical Character Recognition for Brahmi Script Using Geometric Method," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, no. 3-11, pp. 131-136, 2017.

[5] N. Gautam, R. S. Sharma, and G. Hazrati, "Handwriting Recognition of Brahmi Script (an Artefact): Base of PALI Language," in *Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems*, vol. 2: Springer, Cham, 2016, pp. 519-527.

[6] E. Vellingiriraj, M. Balamurugan, and P. Balasubramanie, "Text Analysis and Information Retrieval of Historical Tamil Ancient Documents Using Machine Translation in Image Zoning," *International Journal ofLanguages Literature and Linguistics*, vol. 2, no. 4, pp. 164-168, 2016, doi: 10.18178/IJLLL.2016.2.4.88.

[7] M. W. A. Kesiman, S. Prum, J.-C. Burie, and J.-M. Ogier, "Study on feature extraction methods for character recognition of Balinese script on palm leaf manuscript images," in *23rd International Conference on Pattern Recognition*, Cancun, Mexico, 2016: IEEE, pp. 4017-4022.

[8] U. Pal and B. Chaudhuri, "Indian script character recognition: a survey," *pattern Recognition*, vol. 37, no. 9, pp. 1887-1899, 2004, doi: 10.1016/j.patcog.2004.02.003.

[9] G. Siromoney, R. Chandrasekaran, and M. Chandrasekaran, "Machine recognition of Brahmi script," *IEEE Transactions on Systems Man and Cybernetics*, vol. SMC-13, no. 4, pp. 648-654, 1983, doi: 10.1109/TSMC.1983.6313155.

[10] N. Gautam, S. S. Chai, and M. Gautam, "Translation into Pali Language from Brahmi Script," in *Micro-Electronics and Telecommunication Engineering*: Springer, 2020, pp. 117-124.

[11] A. Soumya and G. H. Kumar, "Recognition of Historical Records using Gabor and Zonal Features," *Signal and Image Processing: An International Journal*, vol. 6, no. 4, pp. 57-69, 2015, doi: 10.5121/sipij.2015.6405.

[12] N. Gautam, S. S. Chai, and J. Jose, "Recognition of Brahmi Words by Using Deep Convolutional Neural Network," *Preprints* 2020, 2020050455 (doi: 10.20944/preprints202005.0455.v1), 2020.

[13] A. P. Singh and A. K. Kushwaha, "Analysis of Segmentation Methods for Brahmi Script," *DESIDOC Journal of Library & Information Technology*, vol. 39, no. 2, pp. 109-116, 2019.

[14] N. Gautam, S. S. Chai, and M. Gautam, "The Dataset for Printed Brahmi Word Recognition," in *Micro-Electronics and Telecommunication Engineering*: Springer, 2020, pp. 125-133.

[15] R. Sarkhel, A. K. Saha, and N. Das, "An enhanced harmony search method for Bangla handwritten character recognition using region sampling," in *2nd International Conference on Recent Trends in Information Systems*, Kolkata, India, 2015: IEEE, pp. 325-330.

[16] S. Bag, G. Harit, and P. Bhowmick, "Recognition of Bangla compound characters using structural decomposition," *Pattern Recognition*, vol. 47, no. 3, pp. 1187-1201, 2014, doi: 10.1016/j.patcog.2013.08.026.

[17] P. P. Roy, A. K. Bhunia, A. Das, P. Dey, and U. Pal, "HMM-based Indic handwritten word recognition using zone segmentation," *Pattern Recognition*, vol. 60, pp. 1057-1075, 2016, doi: 10.1016/j.patcog.2016.04.012.

[18] S. Bhowmik, S. Malakar, R. Sarkar, and M. Nasipuri, "Handwritten Bangla word recognition using elliptical features," in *International Conference on Computational Intelligence and Communication Networks*, Bhopal, India, 2014: IEEE, pp. 257-261, doi: 10.1109/CICN.2014.66.

[19] S. Bhowmik, M. G. Roushan, R. Sarkar, M. Nasipuri, S. Polley, and S. Malakar, "Handwritten

Bangla word recognition using HOG descriptor," in *Fourth International Conference of Emerging Applications of Information Technology*, Kolkata, India, 2014: IEEE, pp. 193-197, doi: 10.1109/EAIT.2014.43.

[20] P. M. Kamble and R. S. Hegadi, "Handwritten Marathi character recognition using R-HOG Feature," *Procedia Computer Science*, vol. 45, pp. 266-274, 2015, doi: 10.1016/j.procs.2015.03.137.

[21] D. Singh, J. Saini, and D. Chauhan, "Hindi character recognition using RBF neural network and directional group feature extraction technique," in *2015 International Conference on Cognitive Computing and Information Processing* Noida, India 2015: IEEE, pp. 1-4.

[22] M. Yadav and R. Purwar, "Hindi handwritten character recognition using multiple classifiers," in *7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*, Noida, India, 2017: IEEE, pp. 149-154.

[23] S. L. Chandure and V. Inamdar, "Performance analysis of handwritten Devnagari and MODI Character Recognition system," in *International Conference on Computing, Analytics and Security Trends*, Pune, India, 2016: IEEE, pp. 513-516, doi: 10.1109/CAST.2016.7915022.

[24] S. Ansari and U. Sutar, "Devanagari Handwritten Character Recognition using Hybrid Features Extraction and Feed Forward Neural Network Classifier (FFNN)," *International Journal of Computer Applications*, vol. 129, no. 7, pp. 22-27, 2015, doi: 10.5120/ijca2015906859.

[25] P. N. Sastry, T. V. Lakshmi, N. K. Rao, T. Rajinikanth, and A. Wahab, "Telugu handwritten character recognition using Zoning features," in *International Conference on IT Convergence and Security*, Beijing, China, 2014: IEEE, pp. 1-4.

[26] C. N. Manisha, E. S. Reddy, and Y. S. Krishna, "Glyph-based recognition of offline handwritten Telugu characters: GBRoOHTC," in *International Conference on Computational Intelligence and Computing Research*, Chennai, India, 2016: IEEE, pp. 1-6, doi: 10.1109/ICCIC.2016.7919567.

[27] P. K. Singh, S. Das, R. Sarkar, and M. Nasipuri, "Recognition of handwritten Indic script numerals using mojette transform," in *Proceedings of the First International Conference on Intelligent Computing and Communication*, Singapore, 2017: Springer, pp. 459-466, doi: 10.1007/978-981-10-2035-3_47.

[28] S. Shreedharamurthy and H. S. Reddy, "Handwritten Kannada numerals recognition using discrete cosine transform coefficients," in *Emerging Research in Electronics, Computer Science and Technology*: Springer, 2014, pp. 909-918.

[29] S. Karthik and K. S. Murthy, "Handwritten Kannada Numerals Recognition Using Histogram of Oriented Gradient Descriptors and Support Vector Machines," in *Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India* 2015: Springer, pp. 51-57.

[30] H. Ashoka, D. Manjaiah, and R. Bera, "Zone based feature extraction and statistical classification technique for Kannada handwritten numeral recognition," *International Journal of Computer Science & Engineering Technology*, vol. 3, no. 10, pp. 476-482, 2012.

[31] M. Padma and S. Pasha, "Quadtree Based Feature Extraction Technique for Recognizing Handwritten Kannada Characters," in *Emerging Research in Electronics, Computer Science and Technology*. New Delhi: Springer, 2014, pp. 725-735.

[32] G. S. G. Rajput and R. Horakeri, "Zone based handwritten Kannada character recognition using crack code and SVM," in *International Conference on Advances in Computing, Communications and Informatics*, Mysore, India, 2013: IEEE, pp. 1817-1821.

[33] M. Alex and S. Das, "An Approach towards Malayalam Handwriting Recognition Using Dissimilar Classifiers," *Procedia Technology*, vol. 25, pp. 224-231, 2016, doi: 10.1016/j.protcy.2016.08.101

[34] P. Gayathri and S. Ayyappan, "Off-line handwritten character recognition using Hidden Markov Model," in *International Conference on Advances in Computing, Communications and Informatics*, New Delhi, India, 2014: IEEE, pp. 518-523.

[35] P. Raj and A. Wahi, "Zone based Method to Classify Isolated Malayalam Handwritten Characters using Hu-Invariant Moments and Neural Networks," *International Innovations in Intelligent Instrumentation, Optimization and Signal Processing*, 2013.

[36] A. T. Jia, Y. Ayappally, and K. Syama, "Malayalam OCR: N-gram approach using SVM classifier," in *International Conference on Advances in Computing, Communications and Informatics*, Mysore, India, 2013: IEEE, pp. 1799-1803, doi: 10.1109/ICACCI.2013.6637454.

[37] K. S. Varghese, A. James, and S. Chandran, "A Novel Tri-Stage Recognition Scheme for Handwritten Malayalam Character Recognition," *Procedia Technology*, vol. 24, pp. 1333-1340, 2016, doi: 10.1016/j.protcy.2016.05.137.

[38] P. Raveena, A. James, and C. Saravanan, "Extended zone based handwritten Malayalam character recognition using structural features," in *Second International Conference on Electrical, Computer and Communication Technologies*, Coimbatore, India, 2017: IEEE, pp. 1-5.

[39] A. M. M. Chacko and P. Dhanya, "A comparative study of different feature extraction techniques for offline Malayalam character recognition," in *Computational Intelligence in Data Mining-Volume 2*: Springer, 2015, pp. 9-18.

[40] G. Singh, C. J. Kumar, and R. Rani, "Performance analysis of different classifiers for recognition of handwritten Gurmukhi characters using hybrid features," in *International Conference on Computing, Communication & Automation*, Noida, India, 2015: IEEE, pp. 1091-1095.

[41] S. J. Macwan and A. N. Vyas, "Classification of offline Gujarati handwritten characters," in *International Conference on Advances in Computing, Communications and Informatics*, Kochi, India, 2015: IEEE, pp. 1535-1541.

[42] M. K. Mahto, K. Bhatia, and R. Sharma, "Combined horizontal and vertical projection feature extraction technique for Gurmukhi handwritten character recognition," in *International Conference on Advances in Computer Engineering and Applications*, Ghaziabad, India, 2015: IEEE, pp. 59-65.

[43] A. A. Desai, "Support vector machine for identification of handwritten Gujarati alphabets using hybrid feature space," *CSI transactions on ICT*, vol. 2, no. 4, pp. 235-241, 2015.

[44] H. R. Thaker and C. Kumbharana, "Structural feature extraction to recognize some of the offline isolated handwritten Gujarati characters using decision tree classifier," *International Journal of Computer Applications*, vol. 99, no. 15, pp. 46-50, 2014, doi: 10.5120/17452-8381.

[45] M. Goswami and S. Mitra, "Structural feature based classification of printed Gujarati characters," in *International Conference on Pattern Recognition and Machine Intelligence*, Kolkata, India, 2013: Springer, pp. 82-87.

[46] A. K. Sharma, D. M. Adhyaru, T. H. Zaveri, and P. B. Thakkar, "Comparative analysis of zoning based methods for Gujarati handwritten numeral recognition," in *5th Nirma University International Conference on Engineering* Ahmedabad, India, 2015: IEEE, pp. 1-5.

[47] P. Solanki and M. Bhatt, "Printed Gujarati script OCR using hopfield neural network," *International Journal of Computer Applications*, vol. 69, no. 13, pp. 33-37, 2013.

[48] C. M. Silva, N. Jayasundere, and C. Kariyawasam, "Contour tracing for isolated Sinhala handwritten character recognition," in *Fifteenth International Conference on Advances in ICT for Emerging Regions*, Colombo, Sri Lanka, 2015: IEEE, pp. 25-31.

[49] K. Dharmapala, W. Wijesooriya, C. Chandrasekara, U. Rathnapriya, and L. Ranathunga, "Sinhala Handwriting Recognition Mechanism Using Zone Based Feature Extraction," presented at the Information Technology Research Unit, Faculty of Information Technology, University of Moratuwa, Sri Lanka, 2017.

[50] H. Naleer and J. Jayamaha, "Feature extraction technique based character recognition using artificial neural network for Sinhala characters," presented at the Enriching the Novel Scientific Research for the Development of the Nation, 2016.

[51] S. Choudhary, T. Patnaik, and S. Singh, "An Efficient Algorithm for Characters recognition of Printed Oriya Script," *Advances in Computer Science and Information Technology*, vol. 2, no. 7, pp. 77-80, 2015.

[52] I. Rushiraj, S. Kundu, and B. Ray, "Handwritten character recognition of Odia script," in *International Conference on Signal Processing, Communication, Power and Embedded System*, Paralakhemundi, India, 2016: IEEE, pp. 764-767.

[53] S. Manjula and R. S. Hegadi, "Recognition of Oriya and English languages based on LBP features," in *Second International Conference on Electrical, Computer and Communication Technologies*, Coimbatore, India, 2017: IEEE, pp. 1-3.

[54] H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Transactions on Electronic Computers*, no. 2, pp. 260-268, 1961.