OFFLINE YORÙBÁ HANDWRITTEN WORD RECOGNITION USING GEOMETRIC FEATURE EXTRACTION AND SUPPORT VECTOR MACHINE CLASSIFIER

Matthias Omotayo Oladele[1], Temilola Morufat Adepoju[1], Olaide Abiodun Olatoke[1] and Oluwaseun Adewale Ojo[1]

[1]Computer Engineering Technology Department,
The Federal Polytechnic, Ede, Nigeria
oladelematthias@gmail.com, atemilola@gmail.com,
laide012002@yahoo.com, oluwaseunojo.2007@gmail.com

ABSTRACT

Yorùbá language is one of the three main languages that is been spoken in Nigeria. It is a tonal language that carries an accent on the vowel alphabets. There are twenty-five (25) alphabets in Yorùbá language with one of the alphabets a digraph (GB). Due to the difficulty in typing handwritten Yorùbá documents, there is a need to develop a handwritten recognition system that can convert the handwritten texts to digital format. This study discusses the offline Yorùbá handwritten word recognition system (OYHWR) that recognizes Yorùbá uppercase alphabets. Handwritten characters and words were obtained from different writers using the paint application and M708 graphics tablets. The characters were used for training and the words were used for testing. Pre-processing was done on the images and the geometric features of the images were extracted using zoning and gradient-based feature extraction. Geometric features are the different line types that form a particular character such as the vertical, horizontal, and diagonal lines. The geometric features used are the number of horizontal lines, number of vertical lines, number of right diagonal lines, number of left diagonal lines, total length of all horizontal lines, total length of all vertical lines, total length of all right slanting lines, total length of all left-slanting lines and the area of the skeleton. The characters are divided into 9 zones and gradient feature extraction was used to extract the horizontal and vertical components and geometric features in each zone. The words were fed into the support vector machine classifier and the performance was evaluated based on recognition accuracy. Support vector machine is a two-class classifier, hence a multiclass SVM classifier least square support vector machine (LSSVM) was used for word recognition. The one vs one strategy and RBF kernel were used and the recognition accuracy obtained from the tested words ranges between 66.7%, 83.3%, 85.7%, 87.5%, and 100%. The low recognition rate for some of the words could be as a result of the similarity in the extracted features.

Keywords: *Gradient, Handwritten, Offline, Pre-processing, Support Vector Machine, Word, Yorùbá, Zoning*

1. Introduction

Handwriting recognition task is a challenging task in machine learning and pattern recognition due to the fact that each individual has his or her unique style of writing which

therefore makes it difficult for the computer to convert it to digital format (Pant *et al.*, 2012). Typing of handwritten text is time-consuming and ineffective (Sepahvand *et al.*, 2017) especially Yorùbá texts because of the accents it carries. Therefore, there is a need to develop an offline Yorùbá word recognition system which will convert the handwritten texts to digital format. Handwriting recognition can be divided into the online and offline recognition system based on the data acquisition or input method. Online recognition converts handwritten text written on a digitizer or PDA automatically where the sensor picks up the pen-tip movements and the pen-up/pen-down switching. The signal obtained from the pen-tip movements is converted into a digital format. Offline recognition converts a text image to a digital format. The image of the written text is scanned and sensed offline by optical scanning or intelligent recognition (Tawde & Kundargi, 2013). In the offline recognition, the fixed static shape of the character is being recognized (Arica & Yarman-Vural, 2001), while the dynamic motion is recognized in online recognition during handwriting. Offline handwriting recognition can be used in many applications such as a bank or check processing, mail sorting, document archiving, commercial form-reading, office automation, and so on (Adeyanju *et al.*, 2014).

Yorùbá language is one of the three major languages spoken in Nigeria. Yorùbá character is tonal thereby making it more difficult than the English language sequence of characters (Ajao *et al.*, 2016). Using under dots make some Yorùbá letters different from English such as in 'Ẹ', 'Ọ', 'Ṣ'. Also, one Yorùbá letter is a digraph that is represented as 'GB'. Like the English alphabets, the Yorùbá alphabets can be divided into vowels and consonants. There are seven vowels (A, E, Ẹ, I, O, Ọ, U) and eighteen consonants (B, D, F, G, GB, H, J, K, L, M, N, P, R, S, Ṣ, T, W, Y) in Yorùbá alphabets making it a total of 25 alphabets, unlike the English alphabet which has 26 alphabets. Yorùbá alphabet does not have the alphabets C, Q, V, X, and Z like we have in English alphabets, but four other alphabets were added i.e. GB (one letter), Ẹ, Ọ, and Ṣ.

Support vector machine (SVM) has been used mostly and effectively in various areas such as pattern recognition, classification, and image processing. It is a supervised learning algorithm that is used for classification and regression. In SVM algorithm, each data item is plotted as a point in n-dimensional space (where n is the number of features extracted) with the value of each feature being the value of a particular coordinate as shown in figure 1. Then, classification is performed by finding the hyper-plane that differentiates the two classes very well (Ray, 2017). Support vectors are the co-ordinates of individual observation. SVM is a linear model that can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is to create a line or a hyperplane which separates the data into classes (Pupale, 2018).
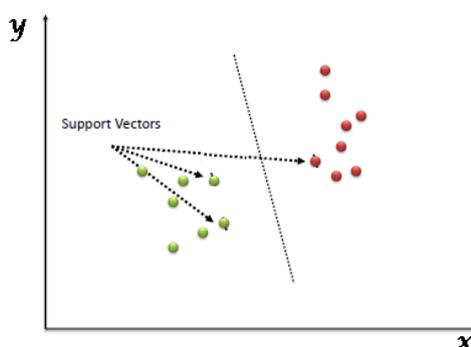

Figure 1: SVM plot with support vectors (Ray, 2017)

Several researches have been done on handwritten character or word recognition, but there is little on Yorùbá recognition, also the few ones focus on character recognition and the research on word recognition were to identify some selected words such as medical words,

etc. because the system was trained with the selected words and as such it is limited to these trained selected words.

Pradeep *et al.* (2011) developed an off-line handwritten alphabetical character recognition system using a multilayer feed-forward neural network, in which diagonal based feature extraction was used to extract the features of the handwritten alphabets. Fifty data sets, each containing 26 alphabets written by various people were used for training the neural network and twenty different handwritten alphabets characters were used for testing. The system shows an accuracy of 96.52% with 54 features and 97.84% with 69 features. Mhetre & Patil (2013) proposed two different methods for numeral recognition and their results were compared. The First method uses a grid-based feature extraction and the feature set is then compared with the feature set of database image for classification, while the second method uses image centroid zone and zone centroid zone algorithms for feature extraction, then the extracted features were then applied to Artificial Neural Network for recognition of input image. Recognition accuracy of 83.60% and 86.40% was obtained from the first and second methods respectively.

Shanjana & James (2015) developed a method to convert handwritten Malayalam documents into a computer recognized digital format. Feature extraction by gradient and curvature computation was used to extract the features and classification was done using the LIBSVM. Overall accuracy of 82% was obtained. Oladele *et al.* (2017) proposed a Yorùbá handwritten character recognition system using support vector machine (SVM). The system recognizes only upper-case characters and achieved a recognition rate of 76.7% and a rejection rate of 23.3%. Ajao *et al.* (2018) developed an offline Yorùbá character recognition system using Freeman chain code and K-Nearest Neighbor (KNN). A recognition accuracy of 87.7% was obtained from the KNN classification algorithm and the Freeman chain code. Sonawane & Shelke (2018) tried to classify handwritten Devanagari characters using transfer learning mechanism with the help of Alexnet. Alexnet which is a convolutional neural network is trained over a dataset of around 16870 samples of 22 consonants of Devanagari script. CNN achieved 94.49% validation accuracy and 95.46% test accuracy.

Fan *et al.* (2019) proposed a Mongolian offline handwriting recognition system. The system consists of three parts namely handwritten image preprocessing, mapping of images to grapheme sequences, and sub-word-based language model (LM) decoding. The performance of the sub-words was evaluated at different levels on the open Mongolian offline handwriting dataset (MHW). The bi-syllable 2-gram LM showed the best performance, with 18.32% and 23.22% word-error rates (WERs) on two test sets. Rahman *et al.* (2019) developed a Bengali character recognition system using geometry-based feature extraction. The geometry-based feature extraction method was used to extract the effective features from the Bengali characters for classification purposes. SVM and Artificial Neural Network (ANN) were used as classifiers on self-generated training and testing data sets which contain 2500 different samples of 50 characters in the Bengali character set and an average recognition rate of 84.56% and 74.47% was obtained using SVM and ANN respectively.

Awni *et al.* (2019) proposed an offline Arabic handwritten word recognition system using deep learning ensemble method. Model averaging technique which is an ensemble learning method was used to train three residual networks (ResNet18) models of the Arabic words. Three distinct optimization techniques were used and the IFN/ENIT (v2.0p1e) database which contains 32,492 handwritten Arabic words of 937 unique Arabic words were used to validate the method. An error rate of 7.21 % was obtained from Adaptive Momentum Estimation (Adam) optimizer, 9.41% for SGD&CLR optimizer, and 11.57% for RMSProp optimizer.

Abdul Rahman *et al.* (2019) developed an OCR used to convert the printed parliamentary reports of Hansard Malaysia which can be used to develop the Malaysian Hansard Corpus (MHC). Four OCR tools were used and compared on ten parliamentary

reports containing 62 pages to evaluate the conversion accuracy and error rate of each conversion tool. The reports were converted to a plain text file (.txt) and they obtained a recognition rate which ranges from 97.01% to 99.17%. Vinothini & Subalalitha (2020) developed an offline handwritten character recognition system for the Tamil language. Convolutional Neural Network (CNN) was used for image preprocessing, feature extraction, and classification. An overall accuracy of 92.4% was obtained.

The motivation of this work is to develop an offline Yorùbá word recognition system using the geometric feature extraction and SVM classifier to convert handwritten Yorùbá text to digital format. Other aspects of this paper are arranged as follows. Section 2 presents the materials and methods of the work. Feature extraction and word recognition were discussed in sections 3 and 4 respectively, Section 5 discusses the performance metrics, and the results and discussion were presented in section 6. Lastly, section 7 concludes the paper.

2. Materials and Methods

The offline Yorùbá handwritten word recognition system (OYHWR) was developed and simulated using MATLAB 2015a. Twenty-five different characters for each Yorùbá alphabets were obtained from different individuals using the Paint application and M708 graphics tablets which was connected to the PC using the Autodesk SketchBook application. Training was done on the character samples by providing the entire character dataset as training data while all the samples of the word were used as the test data. The geometric features were extracted using zoning and gradient descent, the extracted geometric features are the number of horizontal lines, number of vertical lines, number of right diagonal lines, number of left diagonal lines, total length of all horizontal lines, total length of all vertical lines, total length of all right-slanting lines and total length of all left-slanting lines. These extracted features were then fed into the SVM classifier. The stages are divided into four namely, data acquisition, pre-processing, feature extraction, and recognition. The framework is shown in figure 2.
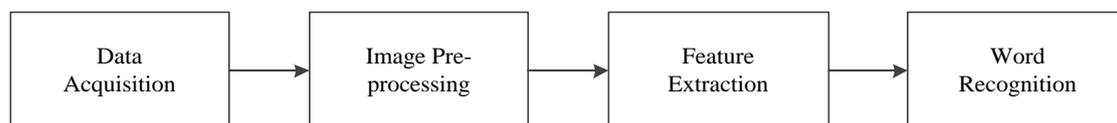

Figure 2: Framework of OYHWR system

2.1. Data Acquisition

A total of 600 Yorùbá characters were used to train the SVM classifier that is 25 each for each character. The word samples were used as the testing set to evaluate the performance of the OYHWR system. The word samples vary from four-letter words to seven-letter words. Figure 3 shows the sample images of the handwritten words.

Figure 3: Sample Handwritten Words

2.2. Image Pre-processing

Image pre-processing is the next step after image acquisition. This stage prepares the acquired images for further processing. Pre-processing helps to enhance the text image such that it will be suitable for feature extraction. Pre-processing that will be performed include grayscale conversion, binarization, noise removal, cropping and resizing, segmentation and skeletonization.

i. **Grayscale Conversion:** This converts the handwritten text to grayscale using the "rgb2gray" function in MATLAB. Grayscale conversion helps to remove hue and saturation information in the sample handwritten image. Equation (1) shows the equation for grayscale conversion.

$$I_{Gray} = 0.298R + 0.587G + 0.114B \tag{1}$$

      Where R, G, and B are the coefficients of Red, Green, and Blue respectively.

ii. **Binarization:** Image binarization takes a grayscale image and converts it to black and white. It helps to reduce the information contained within the image from 0 to 256 to 0 to 1. Binarization is sometimes known as image thresholding. It is a form of segmentation, in which an image is divided into constituent objects. The process of binarization is by finding a threshold value. The equation for binarization is shown in equation (2).

$$g(x, y) = \begin{cases} 1 & if\ f(x,y) \geq T \\ 0 & otherwise \end{cases} \tag{2}$$

      Where T is the Threshold value

iii. **Noise Removal:** Noise removal is the process of removing or reducing the noise from the image. It reduces or removes the visibility of noise by smoothing the entire image leaving areas near contrast boundaries. Morphological operations were used to detect and delete small areas of less than 15px. This stage should be done carefully such that no part of the handwriting is removed. The MATLAB function used is "bwareaopen".

iv. **Cropping and Resizing:** Cropping and resizing extract only the character from the image. The top-leftmost part is the first identified and stored in a temporary variable and also the top-rightmost black pixel, bottom-leftmost, and bottom-rightmost black pixels. After cropping, the image is then resized using the "imresize" function. Resizing allows us to specify the number of rows and columns. The images used in this work were resized to 50×50 pixels.

v. **Segmentation:** Segmentation is an important stage in handwriting recognition. It is used to convert input image consisting of many characters into the individual characters. Segmentation directly affects the recognition accuracy of the system. It is generally performed by dividing single characters from the word image. In this stage, an input image which contains a sequence of characters i.e. words were sub-divided into sub-images of isolated characters. There are several types of segmentation, such as word segmentation and line segmentation. Word segmentation was used in this research since we are dealing with just words and not sentences.

vi. Skeletonization: Skeletonization helps to reduce the foreground regions in a binary image to a skeletal remnant that largely preserves the extent and connectivity of the original region while removing most of the original foreground pixels. Morphological thinning was used which erodes the pixels from the boundary while preserving the endpoints of line segments until no more thinning is possible, at which point what is left is the approximation of the skeleton. It helps to reduce the thickness character image to one-pixel while preserving its connectivity and its topological properties.

3. Feature Extraction

Feature extraction helps to extract the pattern which is most important for the classification. Geometrical features used were number of horizontal lines, number of vertical lines, number of right diagonal lines, number of left diagonal lines, total length of all horizontal lines, total length of all vertical lines, total length of all right slanting lines, total length of all left-slanting lines and the area of the skeleton. These geometrical features were extracted for the individual characters using zoning and gradient. To apply zoning, individual characters were resized into a size of 50 × 50 pixel and thereafter divided into 9 zones i.e. 3 × 3 pixels, then gradient feature extraction was used to extract the geometric features and horizontal and vertical components in each zone. The Gradient computes the magnitude and direction of the greatest change in intensity in a small neighbourhood of each pixel. Gradients are computed through the Sobel operator. The Sobel templates used to compute the horizontal (X) and vertical (Y) components of the gradient are shown in figure 4.

0	1	2
3	4	5
6	7	8

Vertical gradient

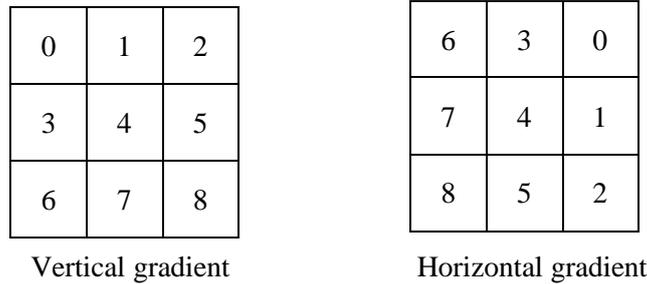| 6	3	0
7	4	1
8	5	2

Horizontal gradient

Figure 4: Sobel mask for vertical and horizontal gradient

For an image, gradient can be calculated by using either the Sobel, Roberts, or Prewitt operator. But the Sobel operator was used to find the gradient vector $\left[G_x, G_y\right]^T$, where G_x is the horizontal gradient component and G_y is the vertical gradient component. Equation (3) and (4) shows the formula for G_x and G_y.

$$G_x(x,y) = I(i-1,j-1) + 2 \times I(i-1,j) + I(i-1,j+1) - I(i+1,j-1) - 2 \times I(i+1,j) - I(i+1,j+1) \quad (3)$$

$$G_y(x,y) = I(i-1,j-1) + 2 \times I(i,j-1) + I(i+1,j-1) - I(i-1,j+1) - 2 \times I(i,j+1) - I(i+1,j+1) \quad (4)$$

Where *I* is the input image with dimensions *M* and *N* at each pixel *(i.j)* where *i = 1 to M* and *j = 1 to N*. The gradient magnitude and direction can be computed from the gradient vector $\left[G_x, G_y\right]^T$ as shown in equation (5) and (6).

$$\text{Gradient Magnitude} \left| G(i,j) \right| = \sqrt{\left(G_x(i,j)\right)^2 + \left(G_y(i,j)\right)^2} \tag{5}$$

$$\theta(i,j) = \tan^{-1} \theta \left\{ \frac{G_y(i,j)}{G_x(i,j)} \right\} \tag{6}$$

The extracted features vectors were then fed into the SVM classifier for training. SVM is a two-class classifier but since word recognition is a multi-class problem LSSVM was adopted. The LSSVM was trained with the training feature sets i.e. the extracted geometrical features from the characters using the one vs one strategy and RBF kernel. The trained SVM classifier was then used during the testing stage for word recognition.

4.   Word Recognition

The final stage of the OYHWR system is word recognition. SVM library known as LSSVM which was run on MATLAB 2015a which was used to classify the handwritten words. Twenty different words ranging from four-letter words to seven-letter words were used to test the classifier. Training was done with the characters before the words were tested. Figure 5 (a) and (b) show the diagram of the training and testing stages. Training is done once, and after the training is complete, the words can be fed into the classifier for testing. The training stage is building the SVM classifier which will be used during the testing stage for word recognition. In the training stage, the character images used for training were pre-processed and the geometrical features were extracted using zoning and gradient feature extraction. The extracted features were trained and the feature vectors were saved for the testing stage. The testing stage tests the handwritten words. The input word image was pre-processed and the features were extracted using zoning and gradient feature extraction. Testing was done by comparing the feature vectors of the test image with the saved feature vectors, the SVM classifier then outputs the predicted result of the word.

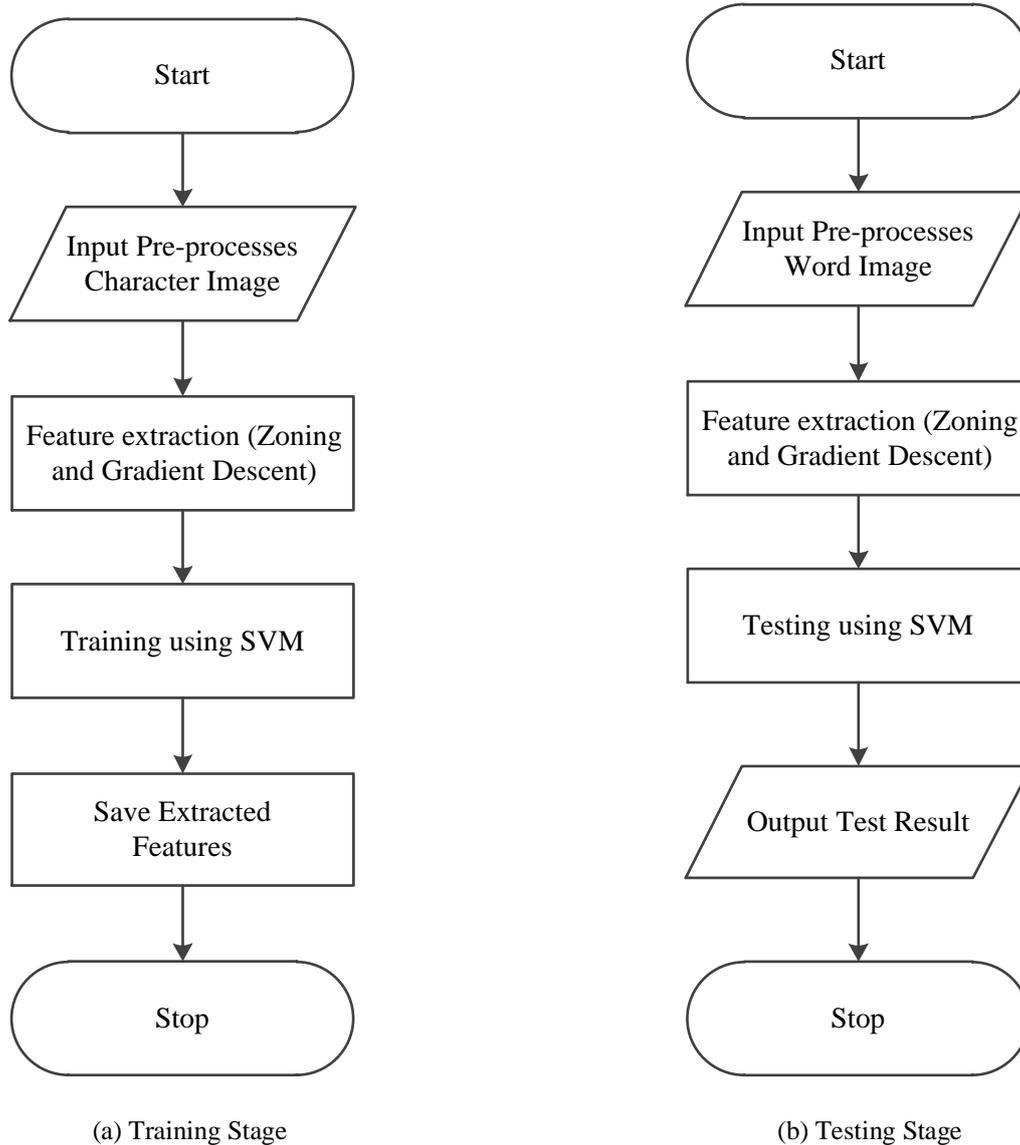(a) Training Stage   (b) Testing Stage

Figure 5: Training and Testing Stages

5.   Performance Evaluation

Performance evaluation helps to measure the performance of a system. The performance evaluation metric used to measure the performance of the developed OYHWR system is recognition accuracy. Recognition accuracy is the ratio of correctly recognised characters in a word to the number of characters in the word. For efficiency, random samples of handwritten words were presented to the model as input and the performance metric was recorded. Equation (7) shows the formula of the recognition accuracy. The recognition accuracy is evaluated in percentage (%).

$$\mathrm{Re}\,cognition\,accuracy = \frac{Number\ of\ correctly\ classified\ characters}{Number\ of\ characters\ in\ the\ word} \tag{7}$$

6.   Results and Discussion

The OYHWR system was tested on an 8GB RAM, Intel core i5, and 2.40GHZ CPU speed HP EliteBook Laptop computer in the MATLAB 2015a environment. Twenty handwritten word samples were presented to the model as input. Table 1 shows the results of some selected words. It was observed from table 1 that a total of 13 words was correctly recognized, while the accuracy of the other 7 words ranges between 66.7%, 83.3%, 85.7%, and 87.5%. The low recognition rate could be as a result of the similarity in the extracted features, for example, the word with the lowest accuracy of 66.7% classified Ẹ as E, and also, there are difficulties in differentiating characters such as A and R, O and Ọ, N and W, G and B.

Table 1: Results of Some Selected Words

Tested Word	Recognized Word	Recognition Rate (%)
ABIRUN	ABIRUN	100
AFỌBAJẸ	AFỌBAJẸ	100
AKINỌLA	AKINỌLA	100
ALAKAN	ALAKAN	100
ALAKỌWE	ALAKỌNE	85.7
BALOGUN	BALỌGUN	85.7
DABIRA	DABIRA	100
ẸDUN	ẸDUN	100
ẸFUN	ẸFUN	100
ẸLẸDẸ	ẸLẸDẸ	100
ELẸGBẸ	ELẸGBE	66.7
ẸLẸSHỌ	ẸLẸSHỌ	100
KABIYESI	KABIYESI	87.5
KOKORO	KỌKỌRỌ	83.3
OGORUN	ỌBỌRUN	83.3
OKỌỌKAN	ỌKỌỌKAN	100
ỌLỌRUN	ỌLỌAUN	83.3
OLUKỌ	OLUKỌ	100
ỌPỌLỌ	ỌPỌLỌ	100
ORIṢA	ORIṢA	100

7.   Conclusion and Future Work

In this study, an offline Yorùbá word recognition (OYHWR) system using SVM was established. The database comprises of six hundred (600) handwritten images with different variation and style which was obtained from several individuals. This variation will help to increase the improve the performance of the system. Yorùbá words and characters were obtained from different individuals. The words and characters were pre-processed and the geometric features were extracted using zoning and gradient feature extraction. The feature vectors were used to train the SVM and the words were fed into the SVM classifier for recognition. The system was evaluated based on recognition accuracy. The recognition rate ranges between 66.7%, 83.3%, 85.7%, 87.5% and 100% when tested with twenty (20) handwritten words. Future work can be done by adding lower case alphabets and numbers and also a standard database for Yorùbá words and characters can be created which can be used to evaluate the performance of any Yorùbá recognition system. Finally, two or more classifiers can be combined to give better recognition accuracy.

References

Abdul Rahman, A. N. C., Abdullah, I. H., Zainuddin, I. S., & Jaludin, A. (2019). The Comparisons of OCR Tools: A Conversion Case in the Malaysian Hansard Corpus Development. *Malaysian Journal of Computing*, 4 (2), 335-348.

Adeyanju, I., Fenwa, O., & Omidiora, E. O. (2014). Effect of non-image features on recognition of handwritten alpha-numeric characters. *International Journal of Computers and Technology*. 2014,13.

Ajao, J. F., Olabiyisi, S. O., Omidiora, E. O., & Okediran, O. O. (2016). Hidden Markov Model Approach for Offline Yorùbá Handwritten Word Recognition. *British Journal of Mathematics & Computer Science*, 18(6), 1-20.

Ajao, J. F., Olawuyi D. O., & Odejobi, O. O. (2018). Yoruba Handwritten Character Recognition using Freeman Chain Code and K-Nearest Neighbor Classifier. *Jurnal Teknologi dan Sistem Komputer*, 6(2), 129-134.

Arica, N., & Yarman-Vural, F. (2001). An Overview of Character Recognition Focused on Off-Line Handwriting. *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, 31(2), 216 - 233.

Awni, M., Khalil, M. I., & Abbas, H. M. (2019). Deep-Learning Ensemble for Offline Arabic Handwritten Words Recognition. *2019 14th International Conference on Computer Engineering and Systems (ICCES)*, Cairo, Egypt. 40 - 45.

Fan, D., Gao, G., & Wu, H. (2019). Sub-Word Based Mongolian Offline Handwriting Recognition. *International Conference on Document Analysis and Recognition (ICDAR)*, Sydney, Australia, 246 - 253.

Mhetre, S. L., & Patil, P. S. (2013). Comparative study of two methods for Handwritten Devanagari Numeral Recognition. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 15(6), 49-53.

Oladele, M. O., Adepoju, T. M., Omidiora, E. O., Sobowale, A. A., Olatoke, O. A., & Ayeleso E. C (2017). An Offline Yorùbá Handwritten Character Recognition System Using Support Vector Machine. *International Conference of Science, Engineering & Environmental Technology (ICONSEET)*, 2(13), 95 - 103.

Pant, A. K., Panday, S. P., & Joshi, S. R. (2012). Off-line Nepali handwritten character recognition using Multilayer perceptron and Radial basis function Neural network, *3rd Asian Himalayas International Conference on Internet*, Kathmandu, Nepal, November 2012.

Pradeep, J., Srinivasan, E., & Himavathi, S. (2011). Diagonal based feature extraction for handwritten character recognition system using neural network. *2011 3rd International Conference on Electronics Computer Technology*, Kanyakumari, 2011, 364 - 368.

Pupale, R. (2018). Support Vector Machines (SVM) - An Overview. *A post at Towards data science available at https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989*

Rahman, M., Mahmud, I., Uddin, P., Afjal, M., Habib, A., & Kabir, F. (2019). Handwritten Bengali Character Recognition Through Geometry-Based Feature Extraction. *Journal of Theoretical and Applied Information Technology,* 97(23), 3570 - 3582.

Ray, S. (2017). Understanding Support Vector Machine (SVM) algorithm from examples (along with code). *A post at Analytics Vidhya available at https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/*

Sepahvand, M., Abdali-Mohammadi, F., & Mardukhi, F. (2017). Evolutionary Metric-Learning Based Recognition Algorithm for Online Isolated Persian/Arabic Characters, Reconstructed Using Inertial Pen Signals. *IEEE Transactions on Cybernetics*, 47(9), 2872 - 2884.

Shanjana, C., & James, A. (2015). Offline Recognition of Malayalam Handwritten Text. In *8th International Conference Interdisciplinarity in Engineering, INTER-ENG 2014, 9th – 10th October 2014, Tirgu-Mures, Romania.* 772-779.

Sonawane, P. K., & Shelke, S. (2018). Handwritten Devanagari Character Classification using Deep Learning. *2018 International Conference on Information, Communication, Engineering, and Technology (ICICET)*, Pune, 1-4.

Tawde, G. Y. & Kundargi, J. M. (2013). An Overview of Feature Extraction Techniques in OCR for Indian Scripts Focused on Offline Handwriting. *International Journal of Engineering Research and Applications (IJERA)*, 3(1).

Vinothini, R., & Subalalitha, C. N. (2020). Offline Handwritten Character Recognition System for Tamil Language. *International Journal of Advanced Science and Technology*, 29(5), 4912 - 4918.