ISSN:0975-9646

A Comparative Analysis of Classifiers Accuracies for Bilingual Printed Documents (Oriya-English)

Sanghamitra Mohanty, Himadri Nandini Das Bebartta
P.G. Department of Computer Science and Application, Utkal University,Vani Vihar, Bhubaneswar, Orissa, India

***Abstract*— Bilingual document recognition has been the subject of intensive research and our focus is on the recognition of an Oriya-English bilingual documents. In most of our official papers, school text books, it is observed that English words interspersed within the Indian languages. So there is need for an Optical Character Recognition (OCR) system which can recognize these bilingual documents and store it for future use. In this paper we present an OCR system developed for the recognition of Indian language i.e. Oriya and Roman scripts for printed documents. For such purpose, it is necessary to separate different scripts before feeding them to their individual OCR system. Firstly, we need to correct the skew followed by segmentation. Here we propose the script differentiation line-wise. We emphasize on Upper and lower matras associated with Oriya and absent in English. We have used horizontal histogram for line distinction belonging to different script. After separation different scripts are sent to their individual recognition engines. Recognition of bilingual script in an image of a document page is of primary importance for a system processing bilingual document. Earlier we had communicated a paper using a single classifier and now three classifiers such as k-nearest neighbor (KNN), convolutional neural networks (CNN) and Support Vector Machines (SVM) schemes have been proposed for analyzing the accuracies for recognition. It has been observed that SVM outperform among all the classifiers.**

***Keywords*— Script separation, Indian script, Bilingual (English-Oriya) OCR, Horizontal profiles, nearest neighbour.**

I. INTRODUCTION

Researchers have been emphasizing a lot of effort for pattern recognition since decades. Amongst the pattern recognition field Optical Character Recognition is the oldest sub field and has almost achieved a lot of success in the case of recognition of Monolingual Scripts. . In India, there are 24 official (Indian constitution accepted) languages. Two or more of these languages may be written in one script. Twelve different scripts are used for writing these languages. Under the three-language formula, some of the Indian documents are written in three languages namely, English, Hindi and the state official language. One of the important tasks in machine learning is the electronic reading of documents. All official documents, magazines and reports can be converted to electronic form using a high performance Optical Character Recognizer (OCR). In the Indian scenario, documents are often bilingual or multi-lingual in nature. English, being the link language in India, is used in most of the important official documents, reports, magazines and technical papers in addition to an Indian language. Monolingual OCRs fail in such contexts and there is a need to extend the operation of current monolingual systems to bilingual ones. This paper describes one such system, which handles both Oriya and Roman script. Recognition of bilingual documents can be approached by the following method i.e. Recognition via script identification. Optical Character Recognition (OCR) system of such a document page can be made through the Development of a script separation scheme to identify different scripts present in the document pages and then run individual OCR developed for each script alphabets. Development of a generalized OCR system for Indian languages is more difficult than a single script OCR development. This is because of the large number of characters in each Indian script alphabet. On the other hand, second option is simpler for a country like India because of many scripts. There are many pieces of work on script identification from a single document. Spitz [1] developed a method to separate Han-based or Latin-based script separation. He used optical density distribution of characters and frequently occurring word shape characteristics for the purpose. Recently, using fractal-based texture features, Tan [5] described an automatic method for identification of Chinese, English, Greek, Russian, Malayalam and Persian text. Ding et al. [3] proposed a method for separating two classes of scripts: European (comprising Roman and Cyrillic scripts) and Oriental (comprising Chinese, Japanese and Korean scripts). Dhanya and Ramakrishnan [8] proposed a Gabor filter based technique for word-wise segmentation from bi-lingual documents containing English and Tamil scripts. Using cluster based templates; an automatic script identification technique has been described by Hochberg et al. [4]. Wood et al. [2] described an approach using filtered pixel projection profiles for script separation. Pal and Chaudhuri [6] proposed a line-wise script identification scheme from tri-language (triplet) documents. Later, Pal et al. [7] proposed a generalized scheme for line-wise script identification from a single document containing all the twelve Indian scripts. Again considering the performance of the system, it largely depends upon the type of classifier used. Because of their parallel processing capabilities as well as learning and decision-making abilities Neural network have been used for pattern recognition since 1950s and neural networks are preferred for pattern recognition problems. The fusion of Neural network and fuzzy systems, termed Neuro-

Fuzzy Systems has also been applied for the solution of various pattern recognition applications. RajaSekaran and Pai[9-11] have investigated the capability of SFAM to behave as a patternRecognizer/Classifier of images for both noisy and noise free using moment based features. G. SVM(Support Vector Machine) classification algorithms, proposed by Vapnik[12] used to solve two-class problem, are based on finding a separation between hyper planes defined. A. Statnikoy, Aliferis and S.Levy in their paper[13] have described multi-category classification methods for micro array gene expression cancer diagnosis using SVM. For multi-class classification, binary SVMs are combined, in either one-against-one (pair wise OVO) scheme or one-against-rest (OVR) or directed acyclic graph (DAGSVM). Due to the high complexity of training and execution, SVM classifiers have been mostly applied to small category set problems. The SVM classifier with RBF Kernel mostly gives highest accuracy.

In the proposed scheme, at first, the documents noise is cleared which we perform at the binarization stage and then the skew is detected and corrected. Using horizontal projection profile the document is segmented into lines. The line height for individual script is different. Along with this property one more uniqueness property in between the Roman and Oriya script is that each line consists of more number of Roman characters as compared to that of Oriya. Basing on these features we have taken a threshold value by dividing the line height of each line with the number of characters in a line. And after obtaining a unique value we sent these lines to their respective classifiers. The classifier which we have used is the Support Vector Machine. The Fig. 1. below shows the entire process carried out for the recognition of our bilingual document.

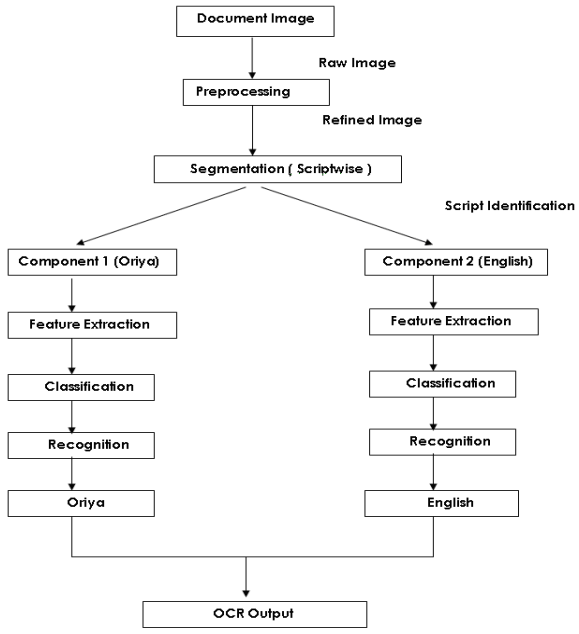


Fig. 1 The schematic representation of the Bilingual OCR system.

In section 2 and Section 3 covers a brief description on properties of Oriya Script and preprocessing techniques. Section 4 gives a description on segmentation. In Section 5 we have described the major portion of our work which focuses on Script identification. Section 6 describes on Feature classification part which has been achieved through Support Vector Machines, Convolutional Neural Network and K-Nearest Neighborhood. Section 8 discusses on the result that we have obtained.

II. PROPERTIES OF ORIYA SCRIPT

As described in the earlier paper on the character set and properties of Oriya Script we come to know that the complex nature of Oriya alphabets consists of 268 symbols (13 vowels, 36 consonants, 10 digits and 210 conjuncts) among which around 90 characters are difficult to recognize because they occupy special size. The character set of Oriya language has been shown in the earlier paper. The components of the characters can be classified into: Main component, Vowel modifier, Consonant Modifier [14].

III. BINARIZATION AND SKEW CORRECTION

The input of an OCR is given in from the scanner or a camera. After this we need to binarize the image. The image enhancement is followed using the spatial domain method that refers to the aggregate of pixels composing an image. Spatial domain processes is denoted by the expression $O(x, y) = T[I(x, y)]$ where $I(x, y)$ is the input image, O(x, y) is the processed image and T is an operator on I. The operator T is applied at each location (x, y) to yield the output. The effect of this transformation would be to produce an image of higher contrast than the original by darkening the levels below 'm' and brightening the levels above m in the original image. Here 'm' is the threshold value taken by us for brightening and darkening the original image. T(r) produces a two-level (binary) image[14].

Detecting the skew of a document image and correcting it are important issues in realizing a practical document reader. For skew correction we have implemented Baird Algorithm. It's a horizontal profiling based algorithm. For skew detection, the horizontal profiles are computed close to the expected orientations. For each angle a measure is made of variation in the bin heights along the profile and the one with the maximum variation gives the Skew angle.

IV. SEGMENTATION

The major challenge in this work is the separation of lines for script identification. The result of line segmentation which has been shown later, takes into consideration the upper and lower *matras* of the line. And this gives the differences in the line height for the distinction of the script. One more factor which we have considered for line identification of different script is the horizontal projection profiles which look into the intensity of pixels

in different zones. Horizontal projection profile is the sum of black pixels along every row of the image. For both of the above methods we have discussed the output in script identification part and here we have discussed the concepts only.

The purpose of analyzing the text line detection of an image is to identify the physical region in the image and their characteristics. A maximal region in an image is the maximal homogenous area of the image. The property of homogeneity in the case of text image refers to the type of region, such as text block, graphic, text line, word, etc. so we define the segmentation as follows

A segmentation of a text line image is a set of mutually exclusive and collectively exhaustive sub regions of the text line image. Given an text line image, I, a segmentation is defined as

$$S = \{R_1, R_2, \cdots, R_n\}, \text{ such that,}$$
$$R_1 \cup R_2 \cup \cdots \cup R_n = I, \text{ and}$$
$$R_i \cap R_j = \emptyset \quad \forall_{i \neq j}.$$

Typical top-down approaches proceed by dividing a text image into smaller regions using the horizontal and vertical projection profiles. The X-Y Cut algorithm, starts dividing a text image into sections based on valleys in their projection profiles. The algorithm repeatedly partitions the image by alternately projecting the regions of the current segmentation on the horizontal and vertical axes. An image is recursively split horizontally and vertically until a final criterion where a split is impossible is met. Projection profile based techniques are extremely sensitive to the skew of the image. Hence extreme care has to be taken while scanning of images or a reliable skew correction algorithm has to be applied before the segmentation process.

V. SCRIPT IDENTIFICATION

In a script, a text line may be partitioned into three zones. The upper-zone denotes the portion above the mean-line, the middle zone (busy-zone) covers the portion of basic (and compound) characters below mean-line and the lower-zone is the portion below base-line. Thus we can define that an imaginary line, where most of the uppermost (lowermost) points of characters of a text line lie, is referred as mean-line (base-line). Example of zoning is shown in Fig. 2. And Fig 3a and b show a word each of Oriya and English, and their corresponding projection profiles respectively. Here mean-line along with base-line partitions the text line into three zones.

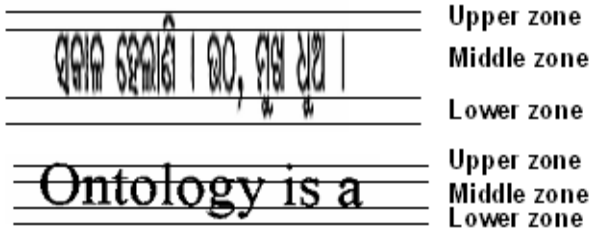


Fig. 2 Line showing the upper, middle and lower zone

For example from the Figure 4 shown below we can observe that the percentage of pixels in the lower zone in case of Oriya characters is more in comparison to English characters.

In this approach, script identification is first performed at the line level and this knowledge is used to identify the OCR to be employed. Individual OCRs have been developed for Oriya [11] as well as English and these could be used for further processing. Such an approach allows the Roman and Oriya characters to be handled independently from each other. In most Indian languages, a text line may be partitioned into three zones. We call the uppermost and lowermost boundary lines of a text line as upper-line and lower-line.

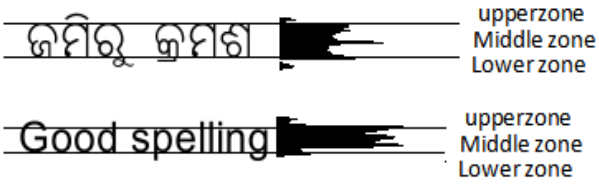


Fig. 3 The three zones of (a) Oriya word and (b) English word

For script recognition, features are identified based on the following observations. From the above projection profile we can observe that

1. The number of Oriya characters present in a line are comparatively less than that of the Roman characters
2. All the upper case letters in Roman script extend into the upper zone and middle zone while the lower case letters occupy the middle, lower and upper zones.
3. The Roman scripts has very few downward extensions(only for g, p, q, j, and y) and have low range of the pixel density, whereas most of the Oriya line contains lower *matras* and have a high range of pixel density.
4. Few Roman scripts(taking into consideration the lower case letters) has very less upward extensions(only for b, d, f, h, l, and t) and have low range of the pixel density, whereas most of the Oriya line contains upper vowel markers *(matras)* and have a high range of pixel density
5. The upper portion of most of the Oriya script is convex in nature and touches the mean-line and the Roman script is dominated by vertical and slant strokes.

In consideration to the above features for distinction we have tried to separate the scripts on the basis of the line height. We cannot emphasize mainly on the density of pixels on the upper and lower zone because if more number of upperward extensions or more number of capital letters are present in upper zone in English character then it is difficult to distinguish that line from Oriya characters. Similarly if more number of downward extensions is present in a line containing English characters and an Oriya line consists of less lower *matras* then also it leads to confusion towards identification. Hence we emphasize the distinction based on line height. Fig 4 shows the different lines extracted for the individual

scripts. Here we have considered the upper and lower *matras* for the Oriya characters. We have observed that, considering a certain threshold value for the line height, document containing English lines have a line height less than the threshold value and the Oriya lines have a value that is greater than the threshold value

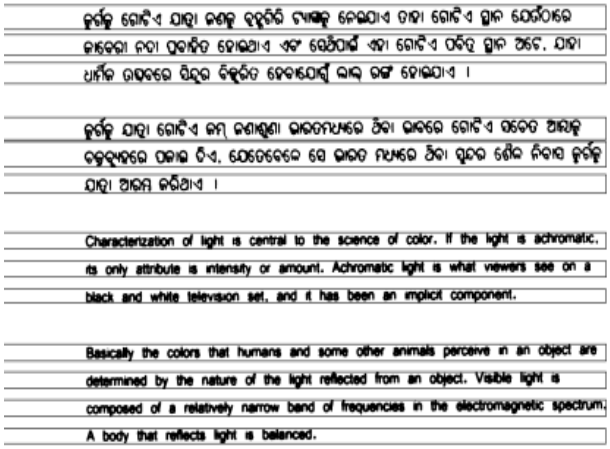


Fig. 4 Shown above is the line with their upper and lower *matra*

TABLE I
LINE HEIGHT FOR THE DIFFERENT LINE NUMBERS

Line Number	Line Height
1	109
2	101
3	98
4	105
5	105
6	72
7	77
8	76
9	71
10	74
11	83
12	77
13	64

TABLE II
THE RATIO OBTAINED AFTER DIVIDING LINE HEIGHT WITH NUMBER OF CHARACTERS

Line Number	Number of Characters
1	3.3
2	3.06
3	3.5
4	3.08
5	3.08
6	8
7	1.08
8	1.08
9	1.24
10	1.08
11	1.25
12	1.08
12	2

For each of the line shown above, the number of characters present in each line has been calculated. Then a threshold value 'R' for both the scripts has been calculated by dividing the line height of each line by the number of characters present in the line. Thus, R can be written as

R = Line height / Number of characters

The values that we obtained have been shown in Table II. From these values we can see that for Oriya script the value lies above 3.0 and for Roman it is below 3.0. So basing on these values the script has been separated.

We have taken nearly fifteen hundred printed documents for having a comparison in between the output and deriving a conclusion. The above table and figure are represented for one of the document while carrying out our experiment.

VI. SCRIPT IDENTIFICATION

The two essential sub-stages of recognition phase are feature extraction and classification. The feature extraction stage analyzes a text segment and selects a set of features that can be used to uniquely identify the text segment. The derived features are then used as input to the character classifier. The classification stage is the main decision making stage of an OCR system and uses the extracted feature as input to identify the text segment according to the preset rules. Performance of the system largely depends upon the type of the classifier used. Classification is usually accomplished by comparing the feature vectors corresponding to the input text/character with the representatives of each character class, using a distance metric. Earlier, we have communicated a paper using Support Vector Machines as a single classifier [16]. In this paper we have tried to show the comparative analysis of the performance of three different classifiers namely K Nearest Neighborhood, Convolutional Neural Network, and Support Vector Machine (SVM).

A. K-Nearest Neighbor:

K nearest neighbor classifier is one among the instance-based methods and it is also called as lazy algorithm. In k-nearest neighbors (k-NN), the posteriori probability of occurrence of unknown pattern is predicted on the basis of frequency of its nearest k-neighbors in a given training sample set. The k-nearest neighbor (k-NN) approach attempts to compute a classification function by examining the labeled training points as nodes or anchor points in the n-dimensional space, where n is feature size. We calculate the Euclidean distance between the test point and all the reference points in order to find K nearest neighbors, and then rank the obtained distances in ascending order and take the reference points corresponding to the k smallest Euclidean distances. A test sample is then attributed the same class label as the label of the majority of its K nearest (reference) neighbors. Euclidean distance is the straight line distance between two points in n-dimensional space. The Euclidean distance between an input feature vector X and a library feature vector C is given by following equation:

$$\sqrt{\sum_{i=1}^{N}(C_i - X_i)(C_i - X_i)}$$

A) K-Nearest Neighbors Rule

To predict the class of a new unlabeled sample u, k samples are searched from the training samples set, which are closest to u and assign u the label of samples that appears the most frequently out of k samples. In another way, assign u the label of samples that appears in majority out of k nearest samples. This is also called as majority rule.There may be some cases where the value of k is even or the value of k>2 and all the k samples belong to different classes leading to ambiguity. In such cases it is essential to break a tie in the number of neighbors. A random and nearest tiebreaker is taken which uses the nearest neighbor among tied groups to break the tie in conflicting situation. There are many methods; those can be used as a metric for computing the distance between training and a test sample. Here Euclidean distance has been used.

Computation time to test a pattern in k-NN depends upon the number of training samples m and the size of feature vector n which is O (n*m). The performance of a classifier further depends upon the value of k, the size of training data set, the metric distance used to measure the distance between a test sample and the training samples and the mode of decisions (majority rule, weighted decision etc.). Various efforts have been made to improve the efficiency of k-NN classifier in respect of computational speed and the classification accuracy

B. Convolutional Neural Network

A convolution network consists of a set of layers each with one or more planes. We can refer to the given Fig 5. Each unit in a plane receives input from a small neighborhood in the planes of the previous layer. The weights of a plane are forced to be equal at all points in the plane and each plane can be considered as a map of a certain feature. Multiple planes are usually used in each layer so that multiple features can be detected. Once a feature has been detected, a sub sampling layer that does a local averaging of the weights follows the convolution layer. Shared weights help reduce the number of parameters.

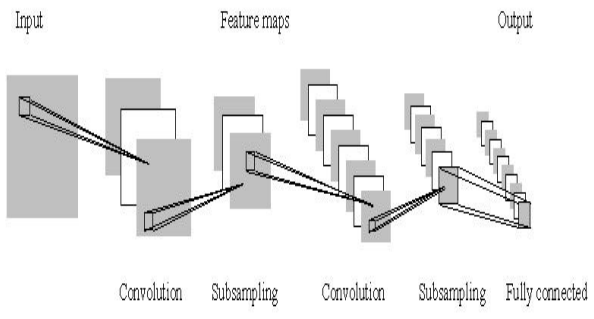


Fig. 5 Convolution neural network with sub sampling layer

We organize the neurons of a given layer into several feature maps. The neurons composing a given feature map will all share weights and common receptive fields. Each such map may, therefore, be viewed as acting to detect a given feature, wherever it occurs. Furthermore, at higher layers of the net, feature maps may be regarded as identifying the combinations of various low-level features that compose more complex higher-level features. These maps represent discrete, localizable detectors for specific features that distinguish among the various classes.

To produce the feature vector for an input document pattern we adopted the two-stage procedure as follows. In the first stage, each of the documents is morphologically dilated in horizontal, vertical, right-diagonal and left-diagonal directions, using 3 x 3 masks. In the next stage, later defined four modified constraint of the image and the original are used for feature extraction. The number of pixels in each of these resulting images is counted.

Our Multi-class Classifier using Convolution Neural Network (C-NN) contains three layers only. These layers are numbered LAYER 0 (convolution layer), LAYER 1 and LAYER 2 (both fully-connected). LAYER 0 is the input layer and its output goes to LAYER 1.Output from LAYER 1 goes to LAYER 2. LAYER 2 is the output layer and the number of neurons in the output layer should normally equal the number of classes. The neuron has information about input matrix, weight matrix, bias input and output matrix.

Each MLP network has 400 neurons in input layer according to the feature column matrix. The number of hidden layer neurons is 400 which are selected by trial and error method. The learning rate is 0.0001. By appropriate selection of these parameters, the Neural Network proved to be robust, strong, and efficient and showed good performance. The network was trained with more than 10,000 Oriya characters by adopting delta learning rule of back propagation algorithm.

In our work we have considered input image in the form of 32*32 and it is fed to the First Layer. The image is normalized and for this we consider a kernel of 3*3 or 5*5. Here we try to reduce the size of the image by preserving its features. So if we consider the image with M rows and N columns, and the kernel has m rows and n columns, then the size of the input image will have M - N + 1 rows, and N – n + 1 columns. From this we can derive that if we take the input image with M value 32 and N value 32 with kernel size having m value 3and n value 3 then in the hidden layer the input size gets reduced to 30*30 according to our mathematical formulae which is shown below. Mathematically we can write the convolution as

$$O(i, j) = \sum_{k=1}^{m} \sum_{l=1}^{n} I(i + k - 1, j + l - 1)\, K(k, l)$$

where I run from 1 to M – m + 1 and j runs from 1 to N – n + 1. We use 2 passes namely as the forward pass and backward pass. In Forward Pass, we calculate output values at each neuron at each layer starting from layer-0 to output layer. And in Backward Pass, a function is used in the learning process, the weights associated with each neurons at each layer gets updated, process starting from output layer continues until layer 0, the errors at output layer are propagated to the layers below it, the weights will get modified according to the learning rule. The back-propagation learning algorithm was chosen adaptive, i.e. changing the learning rate during the training [17, 18].

To avoid local minima of the error function, the batch gradient descent with momentum and variable learning rate [17] was used.

From several experiments we found out that when the recognition percentage is above 90%, the recognition is correct and accurate. Though it is time consuming still it has a good accuracy rate.

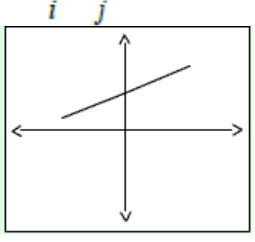
C. Support Vector Machines

SVM (Support Vector Machine) is a useful technique for data classification [19, 20]. The Support Vector Machine (SVM) is learning machine with very good generalization ability, which has been applied widely in pattern recognition, regression estimation, isolated handwritten character recognition, object recognition speaker identification, face detection in images and text categorization. SVM implements the Structural Risk Minimization Principal which seeks to minimize an upper bound of the generalization error. SVM is a kind of learning machine whose fundamental is statistics learning theory. An SVM classifier discriminates two classes of feature vectors by generating hyper-surfaces in the feature space, which are "optimal" in a specific sense that is the hyper-surface obtained by the SVM optimization is guaranteed to have the maximum distance to the "nearest" training examples, the support vectors. On the binary linear separable case, SVM determines the optimal hyper-plane through maximizing the margin between the separating hyper-plane and the data, subjecting to the constraint of classifying the training samples correctly. This can be regarded as an approximate implementation of the structure risk minimization (SRM) principle. SVM can be extended easily to the nonlinear classifier by projecting the data into a high dimensional feature space. It only needs to compute the dot product in the input space rather than in the feature space via constructing a certain kernel function, even need not know the mapped patterns explicitly. SVM operate on kernel evaluations K (P_i, P_j) of the feature vectors P_i or P_j. Variant learning machines are constructed according to the different kernel functions and thus construct different hyper planes in the feature space.

Different types of kernel functions of SVM:

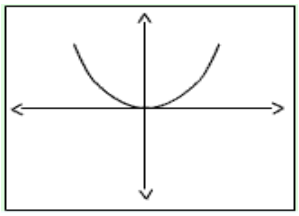
Linear kernel:

$$K(x_i, x_j) = x_i^T x_j$$



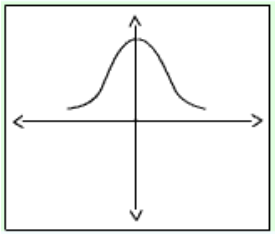
Polynomial Kernel

$$K(x_i, x_j) = (x_i^T x_j + 1)^d$$



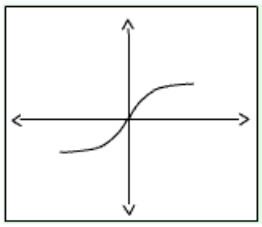
RBF Kernel

$$K(x_i, x_j) = \exp(-\|x_i - x_j\|^2 / 2\sigma^2)$$



Sigmoid Kernel

$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j - \Theta)$$



We have implemented RBF kernel for our work.

VII. RESULTS

A corpus for Oriya OCR consisting of data base for machine printed Oriya characters has been developed. Collection of different samples for both the scripts has been done. Mainly samples have been gathered from laser print documents, books and news papers containing variable font style and sizes. A scanning resolution of 300 dpi is employed for digitization of all the documents. Fig 6 and Fig 7 shows some sample characters of various fonts of both Oriya and Roman script used in the experiment.

We have performed experiments with different types of images such as normal, bold, thin, small, big, etc. having varied sizes of Oriya and Roman characters. The training and testing set comprises of more than 10, 000 samples. We have considered gray scale images for collection of the samples. This database can be utilized for the purpose of document analysis, recognition, and examination. The training set consists of binary images of 297 Oriya letters and 52 English alphabets including both the lower and upper case letters. We have kept the same data file for testing and training for all types of different classifiers to analyze the result. In most of the documents the occurrence of Roman characters is very few as compared to that of Oriya characters. For this reason, for training purpose we have collected more samples for Oriya characters than that of English.

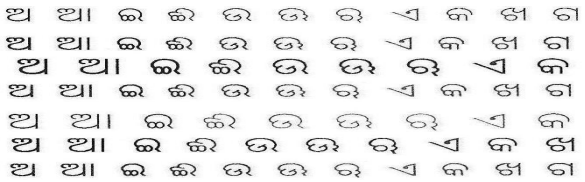


Fig. 6 Samples of machine printed Oriya characters used for training

Fig. 7 Samples of machine printed Roman characters used for training

Regarding the effect on accuracy by considering the different classifiers with different types of the images used for characters, for Oriya-Bold and big characters the accuracy rate is high in case of support vector machines and it has nearly 98.9 percentage of accuracy. The accuracy rate decreases for the thin and small size characters.

TABLE III
CLASSIFIERS ACCURACIES ON ORIYA CHARACTERS

Different types of Classifiers	Accuracy Rate
K- Nearest Neighborhood	96.47
Convolutional Neural Network	96.53
Support Vector Machine	98.9

TABLE IV
CLASSIFIERS ACCURACIES ON ROMAN CHARACTERS

Different types of Classifiers	Accuracy Rate
K- Nearest Neighborhood	95.91
Convolutional Neural Network	95.41
Support Vector Machine	98.43

Table below shows the effect on accuracy for both Oriya and English by considering different character sizes with different types of the images using Support Vector Machines

TABLE V
EFFECT ON ACCURACY BY CONSIDERING DIFFERENT CHARACTER SIZES WITH DIFFERENT TYPES OF THE IMAGES USED FOR ORIYA CHARACTERS.

Image type	Size of the samples	Accuracy percentage
ଅ ଆ ଇ ଈ	Bold and small	92.78%
ଅ ଆ ଇ ଈ	Bold and big	98.9%
ଈ ଓ ଓ ଔ	Normal and small	96.98%
ଅ ଆ ଇ	Normal and Bold	97.12%

Table shows the recognition accuracy for Roman characters with normal, large and bold and small fonts and it is observed that large sizes give better accuracy as compared to the other fonts.

TABLE VI
RECOGNITION ACCURACY FOR ROMAN CHARACTERS WITH DIFFERENT FONT STYLES

Size of the samples	Accuracy percentage
Bold and Big	98.43%
normal	87.78%
Normal and small	88.26%
Normal and Bold	90.89%

VIII. RESULTS

In this paper, the bilingual character recognition capability for Oriya and Roman isolated characters has been discussed. In this work we have tried to distinguish between the English and Oriya documents through the line height and the number of characters present in that line. Separation of the scripts is preferred because training both the scripts in a single recognition system decreases the accuracy rate. There is a probability of some Oriya characters to get confused with some Roman characters and similar problem can be faced during the period of post processing.

We have tried to recognize the Oriya scripts and Roman script with two separate training set using different classifiers. And finally those recognized characters are inserted into a single editor. Classifiers accuracies using different set of features for Oriya and Roman characters are compared. The results obtained are quite encouraging. With NN, recognition accuracy is low as compared to nearest neighbour classifier (KNN) for Oriya and Roman characters. All these results are without applying any post-processing operations. Neural classifiers have much less parameters, and the number of parameters is easy to control. Neural classifiers consume less storage and computation than SVMs. Biggest limitation of the support vector approach lies in the choice of the kernel. Second limitation is speed and size, both in training and testing.

Improved accuracy is always desired, and we are trying to achieve it by improving each and every processing task: pre-processing, feature extraction, sample generation, classifier design, multiple classifier combination, etc. Selection of features and designing of classifiers jointly also lead to better classification performance. Multiple classifiers is being tried to be applied to increase overall accuracy of the OCR system as it is difficult to optimize the performance using single classifier at a time with a larger feature vector set. The present OCR system deals with clean machine printed text with minimum noise. And the input texts are printed in a non italic and non decorative regular font in standard font size. This work in future can be extended for the development of bilingual OCR dealing with degraded, noisy machine printed and italic text. This research work can also be extended to the handwritten text. A postprocessor for both the scripts can also be developed to increase the overall accuracy. So in consideration to the above problems, steps are being taken for more refinement of our bilingual OCR.

ACKNOWLEDGMENT

We are thankful to DIT, MCIT for its support and my colleague Mr. Tarun Kumar Behera for his cooperation.

REFERENCES

[1] A. L. Spitz. "Determination of the Script and Language Content of Document Images". IEEE Trans. On PAMI, pp. 235-245, 1997.

[2] J. Ding, L. Lam,and C. Y. Suen. "Classification of Oriental and European Scripts by using Characteristic Features". In Proceedings of 4th ICDAR, pp. 1023-1027, 1997

[3] D. Hhanya, A. G. Ramakrishna, and P. B. Pati. " Script Identification in Printed Bilingual Documents. Sadhana", 27(1): pp. 73-82, 2002

[4] J. Hochberg, P . Kelly, T. Thomas, and L. Kerns. "Automatic script Identification from Document Images using Cluster-Based Templates" IEEE Trans. on PAMI, pp. 176-181, 1997

[5] T. N. Tan. "Rotation Invariant Texture Features and their use in Automatic Script Identification". IEEE Trans. On PAMI, pp. 751-756, 1998

[6] S. Wood, X. Yao, and K. Krishnamurthi, , L. Dang. "Language Identification for Printed Text Independent of Segmentation". In Proc. Int'l Conf. on Image Processing. Pp. 428-431, 1995

[7] U. Pal, and B. B Chaudhuri,. "Script Line Separation from Indian Multi-Script Documents". IETE Journal of Research, 49, 3-11, 2003

[8] S. Chanda, U. Pal, "English, Devnagari and Urdu Text Identification". Proc. International Conference on Cognition and Recognition, pp. 538-545, 2005

[9] S. Rajasekaran and G. A. V. Pai, "Application of Simplified Fuzzy ARTMAP to structuralengineering problems," All India Seminar on Application of NN in Science, Engineering and Management, Bhubaneswar, June 1997

[10] S. Rajasekaran and G. A. Vijayalakshmi Pai, "Image Recognition using Simplified Fuzzy ARTMAP Augmented with a Moment Based Feature Extractor", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 14, pp. 1081-1095, 2000.

[11] S. Rajasekaran and G. A. Vijayalakshmi Pai, "Simplified Fuzzy ARTMAP As Pattern Recognizer", Journal of Computing in Civil Engineering, Vol. 14, pp. 92-99, 2000.

[12] V. Vapnik, "Statistical Learning Theory, New York, NY, USA Wiley-Interscience, 1998.

[13] A. Statnikov, F. Aliferis and Shawn Levy, "A Comprehensive Evaluation of Multi-Category Classification Methods for Micro Array Gene Expression Cancer Diagnosis" Oxford journals, 2004.

[14] S. Mohanty, H. N. Das Bebartta, and T.K . Behera. "An Efficient Blingual Optical Character Recognition (English-Oriya) System for Printed Documents". Seventh International Conference on Advances in Pattern Recognition, ICAPR. Pp. 398-401, 2009

[15] S. Mohanty, and H. K. Behera." A complete OCR Development System for Oriya Script". Proceedings of SIMPLE' 04, IIT Kharagpur, 2004

[16] S. Mohanty and H. N Das Bebartta, "A Novel Approach for Bilingual (English - Oriya) Script Identification and Recognition in a Printed Documents" International Journal of Image Processing, pp,175-191, 2010.

[17] C. Garcia and M. Delakis, "Convolutional face finder: A neural architecture for fast and robust face detection", IEEE Transaction of Pattern analysis and machine intelligence, vol. 26, 2004

[18] Y. LeCun, L. Bottou, Y. Bengio and. P. Haffner, " Gradient-Based Learning Applied to Document Recognition", Proceedings of the IEEE, vol. 86, no. 11, (1998).

[19] Cao, L.J.; Chong, W.K,"Feature extraction in support vector machine: a comparison of PCA, XPCA and ICA", Proceedings of the 9th International Conference on Neural Information Processing", vol. 2, pp. 1001 – 1005, 2002

[20] Christopher J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", pp.1-43.