A Novel Approach for Bilingual (English - Oriya) Script Identification and Recognition in a Printed Document

Sanghamitra Mohanty sangham1@rediffmail.com
Faculty/Department of Computer Science
Utkal University Bhubaneswar,
751004, India

Himadri Nandini Das Bebartta himadri_nandini@yahoo.co.in
Scholar/ Department of Computer Science
Utkal University Bhubaneswar,
751004, India

Abstract

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.

Keywords: Script separation, Indian script, Bilingual (English-Oriya) OCR, Horizontal profiles

1. 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 [9] 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. Pal et al. [8] also proposed some work on word-wise identification from Indian script documents.

All the above pieces of work are done for script separation from printed documents. 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 Figure 1. below shows the entire process carried out for the recognition of our bilingual document.

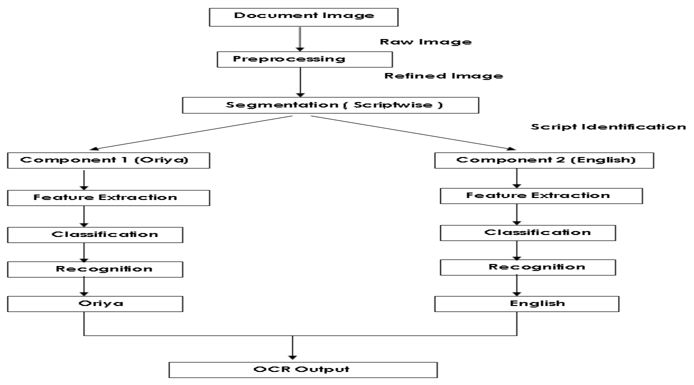


FIGURE 1: The Schematic Representation of the Bilingual OCR system.

In section 2 we have described the properties of the Oriya Script. Section 3 covers a brief description on binarization and skew correction. 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 gives an analysis on the further cases that we have studied for bilingual script differentiation. Section 7 describes on Feature extraction part which has been achieved through Support Vector Machines. Section 8 discusses on the result that we have obtained.

2. PROPERTIES OF ORIYA SCRIPT

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 symbols have been shown in the character set of Oriya language. The components of the characters can be classified into:
(a) Main component: Either a vowel or a symbol may be consonant.
(b) Vowel Modifier: A character can also have a vowel modifier, which modifies the consonants. When the vowel modifier does not touch the main component, it forms separate component, which lies to left or right or top or bottom of the main component and hence does not lie within a line.
(c) Consonant modifier: A symbol can be composed of two or more consonants, the main component and consonant modifier/s or half consonant. Spatially, the consonant modifier could be to bottom or top of the main component, and hence lie above or below the line. More than two up to four consonant vowel combinations are found. These are called conjuncts or *yuktas*. The basic characters of Oriya script are shown in Fig. 2. 1, Fig. 2. 2 and Fig. 2. 3.

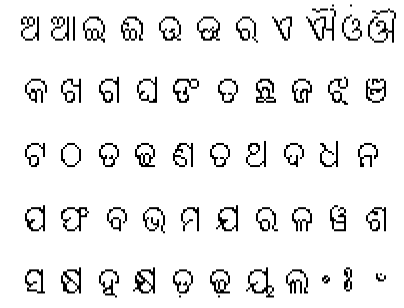


FIGURE 2.1: 52 Oriya Vowels and Consonants.

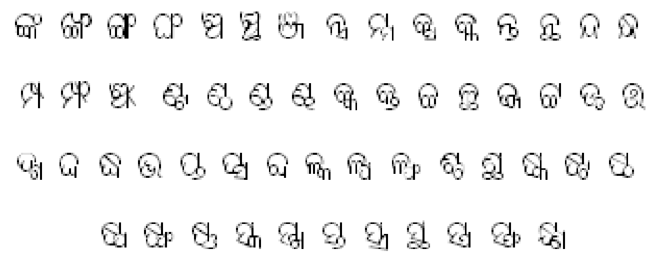


FIGURE 2.2: Some of the Oriya Yuktas (Conjuncts).

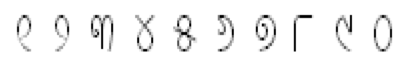


FIGURE 2.3: 10 Oriya Digits.

From the above Figure it can be noted that out of 52 basic characters 37 characters have a convex shape at the upper part. The writing style in the script is from left to right. The concept of upper/lower case is absent in Oriya script. A consonant or vowel following a consonant sometimes takes a compound orthographic shape, which we call as compound character or conjuncts. Compound characters can be combinations of consonant and consonant, as well as consonant and vowel.

3. BINARIZATION AND SKEW CORRECTION

Binarization

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[10].

Skew Correction

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.

4. SEGMENTATION

Several approaches has also been taken for segmentation of the script line wise word wise and character wise. A new algorithm for Segmentation of Handwritten Text in Gurmukhi Script has been done by Sharma and Singh [11]. A new intelligent segmentation technique for functional Magnetic Resonance Imaging (fMRI) been implemented using an Echostate Neural Network (ESN) by D. Suganthi and Dr. S. Purushothaman [12]. A Simple Segmentation Approach for Unconstrained Cursive Handwritten Words in Conjunction with the Neural Network has been performed by Khan and Muhammad [13]. The major challenge in our 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

$$S=\{R_1, R_2, \ldots, R_n\} \text{ ,such that,}$$

$$R_1 \cup R_2 \cup \cdots \cup R_m = I \text{ , and}$$

$$R_i \cap R_j = \phi \;\forall\; i \neq j.$$

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.

5. 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 Figure. 3. And Figure 4a 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.

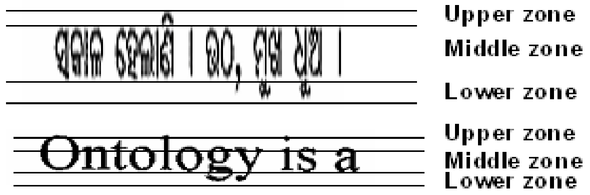


FIGURE 3: Line Showing The Upper, Middle and Lower Zone.

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 [14] 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.

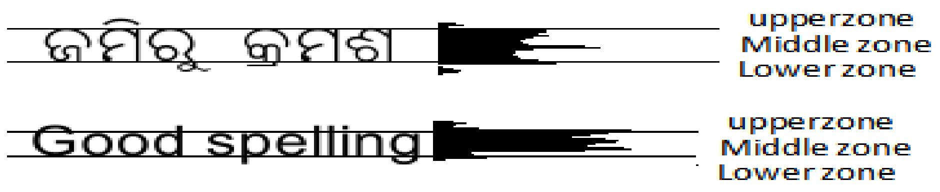
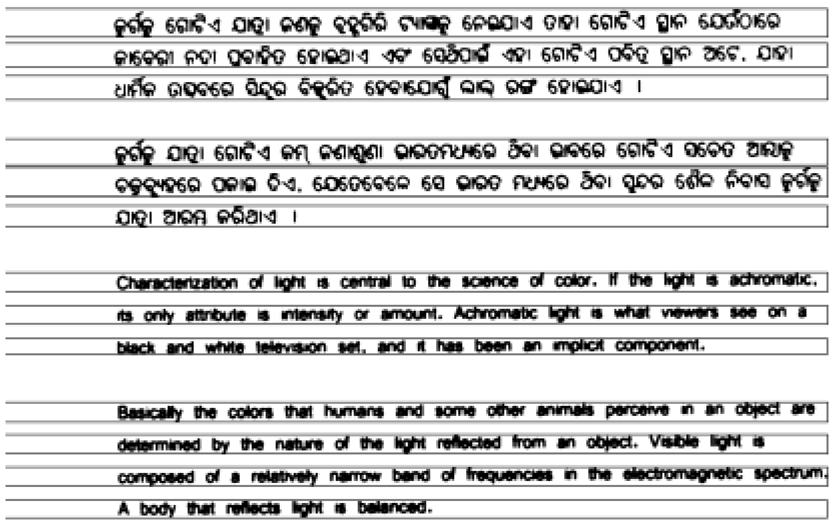


FIGURE 4: The Three Zones of (a) Oriya Word and (b) English Word.

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. Figure 5 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



[illegible]

Characterization of light is central to the science of color. If the light is achromatic,
its only attribute is intensity or amount. Achromatic light is what viewers see on a
black and white television set, and it has been an implicit component.

Basically the colors that humans and some other animals perceive in an object are
determined by the nature of the light reflected from an object. Visible light is
composed of a relatively narrow band of frequencies in the electromagnetic spectrum.
A body that reflects light is balanced.

FIGURE 5: Shown Above is the Line with Their Upper and Lower Matras.

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 1: Line Height for the Different Line Numbers.

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 has been shown in Table 2.From this 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.

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

TABLE 2: The Ratio Obtained after Dividing Line Height with Number of Characters.

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.

6. FEATURE EXTRACTION

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. The classifier which has been used by our system is Support Vector Machine (SVM).

Support Vector Machines

Support vector machines are originally formulated for two-class classification problems [15]. But since decision functions of two-class support vector machines are directly determined to maximize the generalization ability, an extension to multiclass problems is not unique. There are roughly three ways to solve this problem: one against all, pair wise, and all-at-once classifications. Original formulation by Vapnik [15] is one-against-all classification, in which one class is separated from the remaining classes. By this formulation, however, unclassifiable regions exist. Instead of discrete decision functions, Vapnik [16, p. 438] proposed to use continuous decision functions. We classify a datum into the class with the maximum value of the decision functions. In pair wise classification, the n-class problem is converted into $n(n - 1)/2$ two-class problems. Kreßel [18] showed that by this formulation, unclassifiable regions reduce, but still they remain. To resolve unclassifiable regions for pair wise classification, Platt, Cristianini, and Shawe-Taylor [19] proposed decision-tree-based pair wise classification called Decision

Directed Acyclic Graph (DDAG). Kijsirikul and Ussivakul [20] proposed the same method and called it Adaptive Directed Acyclic Graph (ADAG). The problem with DDAGs and ADAGs is that the generalization regions depend on the tree structure [17]. Abe and Inoue [21] extended one-against-all fuzzy support vector machines to pair wise classification. In all-at-once formulation we need to determine all the decision functions at once [22, 23], [16, pp. 437–440]. But this results in simultaneously solving a problem with a larger number of variables than the above mentioned methods. By decision trees, the unclassifiable regions are assigned to the classes associated with leaf nodes. Thus if class pairs with low generalization ability are assigned to the leaf nodes, associated decision functions are used for classification. Thus the classes that are difficult to be separated are classified using the decision functions determined for these classes. As a measure to estimate the generalization ability, we can use any measure that is developed for estimating the generalization ability for two-class problems, because in DDAGs and ADAGs, decision functions for all the class pairs are needed to be determined in advance. We explain two-class support vector machines, pair wise SVMs. and DDAGs.

Two-class Support Vector Machines

Let m-dimensional inputs x_i (i = 1. . . M) belong to Class 1 or 2 and the associated labels be y_i = 1 for Class 1 and –1 for Class 2. Let the decision function be

$$D(x) = W^t x + b$$

where w is an m-dimensional vector, b is a scalar, and

$$y_i \; D(x_i) \geq 1 - \xi_i \quad \text{for} \quad i = 1, \ldots, M.$$

Here ξ_i are nonnegative slack variables. The distance between the separating hyper plane D(x) = 0 and the training datum, with $\xi_i = 0$, nearest to the hyper plane is called margin. The hyper plane D(x) = 0 with the maximum margin is called optimal separating hyper plane. To determine the optimal separating hyper plane, we minimize

$$\frac{1}{2}\|\mathbf{w}\|2 + C\sum_{i=1}^{M}\xi i$$

subject to the constraints:

$$y_i \; (w^t \; x_i \; + b) \geq 1 - \xi_i \quad \text{for} \quad i = 1, \ldots, M.$$

where C is the margin parameter that determines the tradeoff between the maximization of the margin and minimization of the classification error. The data that satisfy the equality in (4) are called support vectors.

To enhance separability, the input space is mapped into the high-dimensional dot-product space called feature space. Let the mapping function be g(x). If the dot product in the feature space is expressed by H(x, x_) =g(x)tg(x), H(x, x_) is called kernel function, and we do not need to explicitly treat the feature space. The kernel functions used in this study are as follows:

1. Dot product kernels

$$H(x, x') = x^t x'$$

2. Polynomial kernels

$$H(x, x') = (x^t x' + 1)^d$$

where d is an integer.

3. Radial Basis Function kernels

$$H(x, x') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$$

where γ value is found out by 1/number of dimension of input vectors.
To simplify notations, in the following we discuss support vector machines with the dot product kernel.

Pair wise Support Vector Machines
In pair wise support vector machines; we determine the decision functions for all the combinations of class pairs. In determining a decision function for a class pair, we use the training data for the corresponding two classes. Thus, in each training the number of training data is reduced considerably compared to one-against-all support vector machines, which use all the training data. But the number of decision functions is n(n – 1)/2, compared to n for one-against-all support vector machines, where n is the number of classes [25].

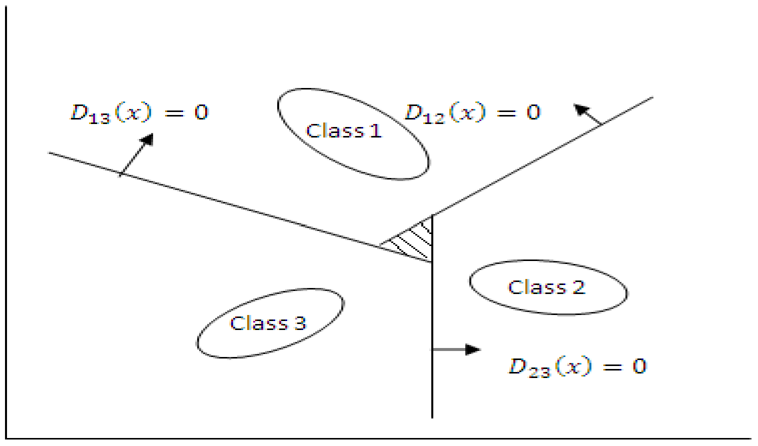


FIGURE 6. Unclassifiable Regions by the Pair Wise Formulation.

Let the decision function for class i against class j, with the maximum margin, be

$$D_{ij}(\mathbf{x}) = w_{ij}^{t}\,\mathbf{x} + b_{ij}$$

where w_{ij} is an m-dimensional vector, b_{ij} is a scalar, and $D_{ij}(x) = -D_{ji}(x)$. The

$$R_i = \{\mathbf{x} \mid D_{ij}(\mathbf{x}) > 0, j = 1, \ldots\ldots, n, j \neq i\}$$

regions do not overlap and if x is in R_i, we classify x into class i. If x is not in R_i ($i = 1, \ldots, n$), we classify x by voting. Namely, for the input vector x we calculate

$$D_i(\mathbf{X}) = \sum_{j=1, j\neq i}^{n} sign(D_{ij}(\mathbf{X}))$$

where

$$\text{sign}(x) = \begin{cases} 1 & for\ x \geq 0 \\ -1 & for\ x < 0 \end{cases}$$

and classify x into the class

$$\arg \max_{i=1,\ldots,n} D_i(x)$$

If $x \varepsilon R_i$, $D_i(x) = n - 1$ and $D_k(x) < n - 1$. Thus x is classified into class i. But if any of $D_i(x)$ is not n, (12) may be satisfied for plural i's. In this case, x is unclassifiable. If the decision functions for a three- class problem are as shown in Figure 6, the shaded region is unclassifiable since $D_i(x) = 1$ ($i = 1, 2,$ and 3).

Decision-tree Based Support Vector Machines:
Decision Directed Acyclic Graphs
Figure 7 shows the decision tree for the three classes shown in Figure 6. In the

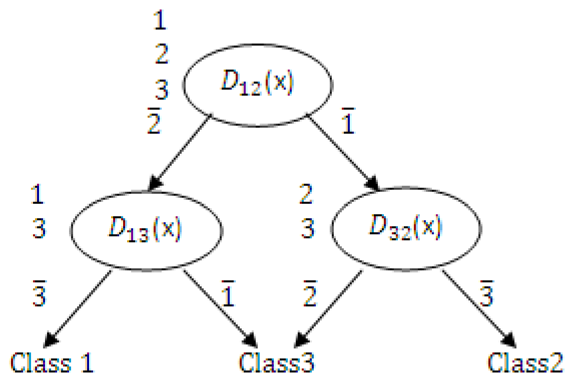


FIGURE 7: Decision-Tree-Based Pair Wise Classification.

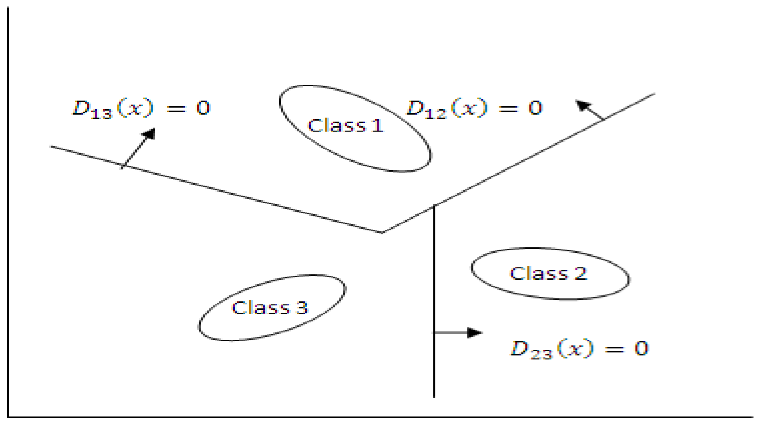


FIGURE 8: Generalization Region by Decision Tree Based Pair Wise Classification.

Figure 7, $\bar{i}$ shows that x does not belong to class i. As the top-level classification, we can choose any pair of classes. And except for the leaf node if D_{ij} (x) > 0, we consider that x does not belong to class j, and if D_{ij} (x) < 0 not class i. Then if D_{12} (x) > 0, x does not belong to Class 2. Thus it belongs to either Class 1 or 3 and the next classification pair is Classes 1 and 3. The generalization regions become as shown in Figure 8. Unclassifiable regions are resolved but clearly the generalization regions depend on the tree structure.

Classification by a DDAG is executed by list processing. In list processing, first we generate a list with class numbers as elements. Then, we calculate the decision function, for the input x, corresponding to the first and the last elements. Let these classes be i and j and D_{ij} (x) > 0. We delete the element j from the list. We repeat the above procedure until one element is left. Then we classify x into the class that corresponds to the element number. For Fig. 7, we generate the list {1, 3, and 2}. If D_{12} (x) > 0, we delete element 2 from the list; we obtain {1, 3}. Then if D_{13} (x) > 0, we delete element 3 from the list. Since only 1 is left in the list, we classify x into Class 1.

Training of a DDAG is the same with conventional pair wise support vector machines. Namely, we need to determine n (n – 1)/2 decision functions for an n-class problem. The advantage of DDAGs is that classification is faster than conventional pair wise support vector machines or pair wise fuzzy support vector machines. In a DDAG, classification can be done by calculating (n – 1) decision functions [24].

We have made use of DDAG support vector machine for the recognition of our OCR engine. And below we show types of samples used for training and testing and the accuracy rate which we have obtained for training characters.

7. 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. Figure 9 and Figure 10 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.

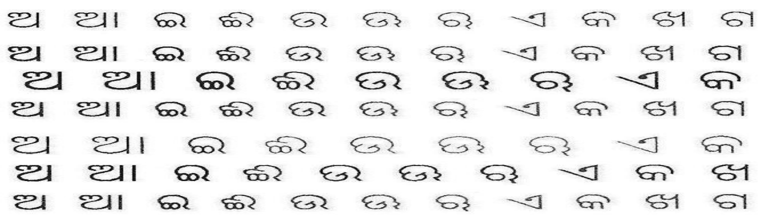


FIGURE 9: Samples of machine printed Oriya Characters Used for Training.

a b c d e f g h I j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h I j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
a b c d e f g h I j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

FIGURE 10: Samples of Machine Printed Roman Characters Used For Training.

Table below shows the effect on accuracy by considering different character sizes with different types of the images used for Oriya characters.

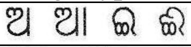
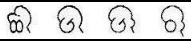
Image type	Size of the samples	Accuracy percentage
	Bold and small	92.78%
	Bold and big	99.8%
	Normal and small	96.98%
	Normal and Bold	97.12%

TABLE 2: Effect on Accuracy by Considering Different Character Sizes with Different Types of the Images used for Oriya Characters.

Table 3 below 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.

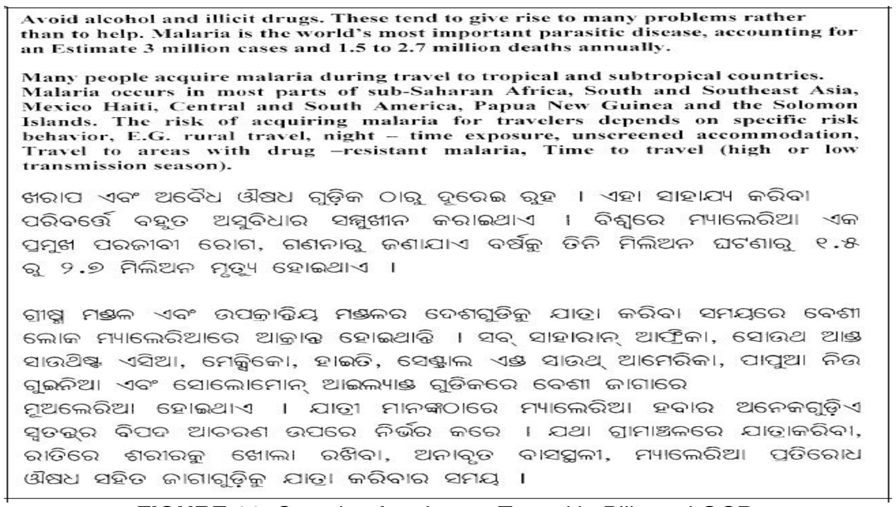
Size of the samples	Accuracy percentage
Large	92.13%
normal	87.78%
Normal and small	88.26%
Normal and Bold	90.89%

TABLE 3: Recognition Accuracy for Roman Characters with Different Font Styles

Regarding the effect on accuracy by considering the different character sizes with different types of the images used for characters, for Oriya-Bold and big characters the accuracy rate is high and it is nearly 99.8 percentage of accuracy. The accuracy rate decreases for the thin and small size characters.

Figure 11 shows an example of a typical bilingual document used in our work. It can be seen that as per our discussion in the script identification section, all most all of the Oriya lines are associated with lower and upper *matras*.

Thus finally after the different scripts being sent to the respective classifiers the final result that we got is shown below in Figure 12. The Figure 11 shows one of the images we have taken for testing. The upper portion of the image contains the English scripts and the lower half of the image consists of the Oriya script.


Avoid alcohol and illicit drugs. These tend to give rise to many problems rather than to help. Malaria is the world's most important parasitic disease, accounting for an Estimate 3 million cases and 1.5 to 2.7 million deaths annually.

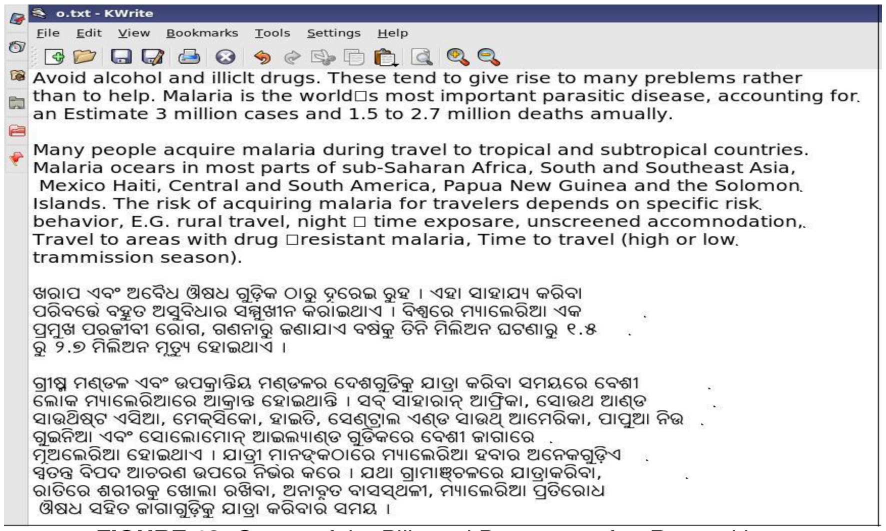
Many people acquire malaria during travel to tropical and subtropical countries. Malaria occurs in most parts of sub-Saharan Africa, South and Southeast Asia, Mexico Haiti, Central and South America, Papua New Guinea and the Solomon Islands. The risk of acquiring malaria for travelers depends on specific risk behavior, E.G. rural travel, night – time exposure, unscreened accommodation, Travel to areas with drug –resistant malaria, Time to travel (high or low transmission season).

ଖରାପ ଏବଂ ଅବୈଧ ଔଷଧ ଗୁଡ଼ିକ ଠାରୁ ଦୂରେଇ ରୁହ । ଏହା ସାହାଯ୍ୟ କରିବା ପରିବର୍ତ୍ତେ ବହୁତ ଅସୁବିଧାର ସମ୍ମୁଖୀନ କରାଇଥାଏ । ବିଶ୍ୱରେ ମ୍ୟାଲେରିଆ ଏକ ପ୍ରମୁଖ ପରଜୀବୀ ରୋଗ, ଗଣନାରୁ ଜଣାଯାଏ ବର୍ଷକୁ ତିନି ମିଲିଅନ ଘଟଣାରୁ ୧.୫ ରୁ ୨.୭ ମିଲିଅନ ମୃତ୍ୟୁ ହୋଇଥାଏ ।

ଗ୍ରୀଷ୍ମ ମଣ୍ଡଳ ଏବଂ ଉପକ୍ରାନ୍ତିୟ ମଣ୍ଡଳର ଦେଶଗୁଡିକୁ ଯାତ୍ରା କରିବା ସମୟରେ ବେଶୀ ଲୋକ ମ୍ୟାଲେରିଆରେ ଆକ୍ରାନ୍ତ ହୋଇଥାନ୍ତି । ସବ୍ ସାହାରାନ୍ ଆଫ୍ରିକା, ସାଉଥ ଆଣ୍ଡ ସାଉଥିଷ୍ଟ ଏସିଆ, ମେକ୍ସିକୋ, ହାଇତି, ସେଣ୍ଟ୍ରାଲ ଏଣ୍ଡ ସାଉଥ୍ ଆମେରିକା, ପାପୁଆ ନିଉ ଗୁଇନିଆ ଏବଂ ସୋଲୋମୋନ୍ ଆଇଲ୍ୟାଣ୍ଡ ଗୁଡିକରେ ବେଶୀ ଜାଗାରେ
ମୂଅଲେରିଆ ହୋଇଥାଏ । ଯାତ୍ରୀ ମାନଙ୍କଠାରେ ମ୍ୟାଲେରିଆ ହବାର ଅନେକଗୁଡ଼ିଏ ସ୍ୱତନ୍ତ୍ର ବିପଦ ଆଚରଣ ଉପରେ ନିର୍ଭର କରେ । ଯଥା ଗ୍ରାମାଞ୍ଚଳରେ ଯାତ୍ରାକରିବା, ରାତିରେ ଶରୀରକୁ ଖୋଲା ରଖିବା, ଅନାବୃତ ବାସସ୍ଥଳୀ, ମ୍ୟାଲେରିଆ ପ୍ରତିରୋଧ ଔଷଧ ସହିତ ଜାଗାଗୁଡ଼ିକୁ ଯାତ୍ରା କରିବାର ସମୟ ।

FIGURE 11: Sample of an Image Tested in Bilingual OCR.

For the image shown above the corresponding output is shown below in Figure 12.



```
Avoid alcohol and illiclt drugs. These tend to give rise to many preblems rather
than to help. Malaria is the world□s most important parasitic disease, accounting for
an Estimate 3 million cases and 1.5 to 2.7 million deaths amually.

Many people acquire malaria during travel to tropical and subtropical countries.
Malaria ocears in most parts of sub-Saharan Africa, South and Southeast Asia,
 Mexico Haiti, Central and South America, Papua New Guinea and the Solomon
Islands. The risk of acquiring malaria for travelers depends on specific risk
behavior, E.G. rural travel, night □ time exposare, unscreened accomnodation,
Travel to areas with drug □resistant malaria, Time to travel (high or low
trammission season).

ଖରାପ ଏବଂ ଅବୈଧ ଔଷଧ ଗୁଡ଼ିକ ଠାରୁ ଦୂରେଇ ରୁହ । ଏହା ସାହାଯ୍ୟ କରିବା
ପରିବର୍ତ୍ତେ ବହୁତ ଅସୁବିଧାର ସମ୍ମୁଖୀନ କରାଇଥାଏ । ବିଶ୍ୱରେ ମ୍ୟାଲେରିଆ ଏକ
ପ୍ରମୁଖ ପରଜୀବୀ ରୋଗ, ଗଣନାରୁ ଜଣାଯାଏ ବର୍ଷକୁ ତିନି ମିଲିଅନ ଘଟଣାରୁ ୧.୫
ରୁ ୨.୭ ମିଲିଅନ ମୃତ୍ୟୁ ହୋଇଥାଏ ।

ଗ୍ରୀଷ୍ମ ମଣ୍ଡଳ ଏବଂ ଉପକ୍ରାନ୍ତିୟ ମଣ୍ଡଳର ଦେଶଗୁଡିକୁ ଯାତ୍ରା କରିବା ସମୟରେ ବେଶୀ
ଲୋକ ମ୍ୟାଲେରିଆରେ ଆକ୍ରାନ୍ତ ହୋଇଥାନ୍ତି । ସବ୍ ସାହାରାନ୍ ଆଫ୍ରିକା, ସୋଉଥ ଆଣ୍ଡ
ସାଉଥିଷ୍ଟ ଏସିଆ, ମେକ୍ସିକୋ, ହାଇତି, ସେଣ୍ଟ୍ରାଲ ଏଣ୍ଡ ସାଉଥ୍ ଆମେରିକା, ପାପୁଆ ନିଉ
ଗୁଇନିଆ ଏବଂ ସୋଲୋମୋନ୍ ଆଇଲ୍ୟାଣ୍ଡ ଗୁଡିକରେ ବେଶୀ ଜାଗାରେ
ମ୍ଅଲେରିଆ ହୋଇଥାଏ । ଯାତ୍ରୀ ମାନଙ୍କଠାରେ ମ୍ୟାଲେରିଆ ହବାର ଅନେକଗୁଡ଼ିଏ
ସ୍ୱତନ୍ତ୍ର ବିପଦ ଆଚରଣ ଉପରେ ନିର୍ଭର କରେ । ଯଥା ଗ୍ରାମାଞ୍ଚଳରେ ଯାତ୍ରାକରିବା,
ରାତିରେ ଶରୀରକୁ ଖୋଲା ରଖିବା, ଅନାବୃତ ବାସସ୍ଥଳୀ, ମ୍ୟାଲେରିଆ ପ୍ରତିରୋଧ
 ଔଷଧ ସହିତ ଜାଗାଗୁଡ଼ିକୁ ଯାତ୍ରା କରିବାର ସମୟ ।
```

FIGURE 12: Output of the Bilingual Document after Recognition.

8. CONCLUSION

A novel method to script separation has been taken care of in this paper. In this work we have tried to distinguish between the English and Oriya documents through horizontal projection profiles for intensity of pixels in different zone along with 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 Support Vector Machines. And finally those recognized characters are inserted into a single editor. Improved accuracy is always desired, and we are trying to achieve it by improving each and every processing task: preprocessing, 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.

ACKNOWLEDGEMENT

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

9. REFERENCES

1. A. L. Spitz. “Determination of the Script and Language Content of Document Images”. IEEE Trans. on PAMI, 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): 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, 176-181, 1997

5. T. N. Tan. “Rotation Invariant Texture Features and their use in Automatic Script Identification”. IEEE Trans. On PAMI, 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. 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. U. Pal, S. Sinha, and B. B. Chaudhuri. “Multi-Script Line identification from Indian Documents”. In Proceedings 7th ICDAR, 880--884, 2003

9. S. Chanda, U. Pal, “English, Devnagari and Urdu Text Identification”. Proc. International Conference on Cognition and Recognition, 538-545, 2005

10. 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. 398-401, 2009

11. R. K. Sharma, Dr. A. Singh, “Segmentation of Handwritten Text in Gurmukhi Script”. Computers & Security, 2(3):12-17, 2009

12. D. Suganthi, Dr. S. Purushothaman, “fMRI Segmentation Using Echo State Neural Network”. Computers & Security, 2(1):1-9, 2009

13. A. R. Khan, D. Muhammad, “A Simple Segmentation Approach for Unconstrained Cursive Handwritten Words in Conjunction with the Neural Network”. Computers & Security, 2(3):29-35, 2009

14. S. Mohanty, and H. K. Behera.” A complete OCR Development System for Oriya Script”. Proceedings of SIMPLE’ 04, IIT Kharagpur, 2004

15. B. V. Dasarathy. “Nearest Neighbor Pattern Classification Techniques”. IEEE Computer Society Press,New York, 1991

16. V. N. Vapnik. “The Nature of Statistical LearningTheory”. Springer-Verlag, London, UK, 1995.

17. V. N. Vapnik. “Statistical Learning Theory”. John Wiley & Sons, New York, 1998.

18. S. Abe. “Analysis of multiclass support vector machines”. In Proceedings of International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA’2003), Vienna, Austria, 2003

19. U. H.-G. Kreßel. “Pair wise classification and support vector machines”. In B. Sch¨olkopf, C. J. C. Burges, and A. J. Smola, editors, Advances in Kernel Methods: Support Vector Learning, pages 255– 268. The MIT Press, Cambridge, MA, 1999

20. J. C. Platt, N. Cristianini, and J. Shawe-Taylor. “Large margin DAGs for multiclass classification”. In S. A. Solla, T. K. Leen, and K.-R. M¨uller, editors, Advances in Neural Information Processing Systems12, pages 547–553. The MIT Press, Cambridge, MA, 2000

21. B. Kijsirikul and N. Ussivakul. “Multiclass support vector machines using adaptive directed acyclic Graph”. In Proceedings of International Joint Conference on Neural Networks (IJCNN 2002), 980–985, 2002

22. S. Abe and T. Inoue. “Fuzzy support vector machines for multiclass problems”. In Proceedings of the Tenth European Symposium on Artificial Neural Networks (ESANN”2002), 116–118, Bruges, Belgium, 2002

23. K. P. Bennett. Combining support vector and mathematical programming methods for classification. In B. Sch¨olkopf, C. J. C. Burges, and A. J. Smola, editors, Advances in Kernel Methods: Support Vector Learning, pages 307–326. The MIT Press, Cambridge, MA, 1999

24. J. Weston and C. Watkins. Support vector machinesfor multi-class pattern recognition. In Proceedings of the Seventh European Symposium on Artificial Neural Networks (ESANN’99), pages 219–224, 1999

25. F. Takahashi and S. Abe. “Optimizing Directed Acyclic Graph Support vector Machines”. ANNPR , Florence (Italy), September 2003