


Article

A Study of Different Classifier Combination Approaches for Handwritten *Indic* Script Recognition

Anirban Mukhopadhyay *, Pawan Kumar Singh , Ram Sarkar * and Mita Nasipuri

Department of Computer Science and Engineering, Jadavpur University, Kolkata-700032, West Bengal, India; pawansingh.ju@gmail.com (P.K.S.); mitanasipuri@gmail.com (M.N.)

* Correspondence: anirbanmcse@gmail.com (A.M.); raamsarkar@gmail.com (R.S.)

Received: 15 December 2017; Accepted: 8 February 2018; Published: 13 February 2018

Abstract: Script identification is an essential step in document image processing especially when the environment is multi-script/multilingual. Till date researchers have developed several methods for the said problem. For this kind of complex pattern recognition problem, it is always difficult to decide which classifier would be the best choice. Moreover, it is also true that different classifiers offer complementary information about the patterns to be classified. Therefore, combining classifiers, in an intelligent way, can be beneficial compared to using any single classifier. Keeping these facts in mind, in this paper, information provided by one shape based and two texture based features are combined using classifier combination techniques for script recognition (word-level) purpose from the handwritten document images. *CMATERdb*8.4.1 contains 7200 handwritten word samples belonging to 12 *Indic* scripts (600 per script) and the database is made freely available at <https://code.google.com/p/cmaterdb/>. The word samples from the mentioned database are classified based on the confidence scores provided by Multi-Layer Perceptron (MLP) classifier. Major classifier combination techniques including majority voting, Borda count, sum rule, product rule, max rule, Dempster-Shafer (DS) rule of combination and secondary classifiers are evaluated for this pattern recognition problem. Maximum accuracy of 98.45% is achieved with an improvement of 7% over the best performing individual classifier being reported on the validation set.

Keywords: Classifier combination; Dempster-Shafer theory of evidence; *Indic* script identification; Histograms of Oriented Gradients; Modified Log-Gabor filter transform; Elliptical features

1. Introduction

In the domain of document images processing, Optical Character Recognition (OCR) systems are, in general, developed keeping a particular script in mind, which implies that such systems can read characters written in a specific script only. This is because the number of characters, shape of the characters or the writing style of using a particular character set is so different that designing a common feature set applicable for recognizing any character set is practically impossible. As an alternative, a pool of OCR systems that correspond to different scripts [1] can be used to solve this said problem. This statement infers that before the document images are fed to an OCR system, it is required to identify the script in which the document is written so that those document images can be suitably converted into a computer-editable format using that OCR system. This summarizes the problem of script identification. There are some important applications of script identification system such as automatic archiving as well as indexing of multi-script documents, searching required information from digitized archives of multi-scripts document images.

In this paper, script identification from handwritten document images written in different scripts is considered. In this regard, it is to be noted that hurdles are multi-fold when handwritten document images are considered compared to its printed counterpart. The main difficulty which researchers

need to deal with is the non-uniformity of the shape and size of the characters written by different writers. Along with these, problems like skew, slant etc. are commonly seen in handwritten documents. Even the paper and ink qualities make things much difficult. Apart from the intrinsic complexities of handwritings, similarities among the characters belonging to different script augment the challenges of script recognition from the handwritten document images. It is worth mentioning that, usually, script recognition is performed at page, text-line or at word level. But in this paper, this is done at word-level because of two reasons: (a) feature extraction at word-level is less time consuming than at page or at text-line level and (b) sometimes, it is seen that a single document page or a single text line contains multiple scripts. In that case, word-level script identification is appropriate.

Script recognition articles for handwritten documents are relatively limited in comparison to its printed counterpart. Ubul et al. [2] comprehensively showed the state-of-the-art performance results for different identification, feature extraction and classification methodologies involved in the process. Recently, Singh et al. [1] provided a survey considering various feature extraction and classification techniques associated with the offline script identification of the *Indic* scripts. Spitz [3] proposed a method for distinguishing between *Asian* and *European* languages by analysing the connected components. Tan et al. [4] developed a method based on texture analysis for automatic script identification from document images using multiple channel (Gabor) filters and Gray level co-occurrence matrices (GLCM) for seven languages: *Chinese, English, Greek, Koreans, Malayalam, Persian* and *Russian*. Hochberg et al. [5,6] described an algorithm for script and language identification from handwritten document images using statistical features based on connected component analysis. Wood et al. [7] demonstrated a projection profile method to determine *Roman, Russian, Arabic, Korean* and *Chinese* characters. Chaudhuri et al. [8] discussed an OCR system to read two Indian languages *viz., Bangla* and *Devanagari (Hindi)*. Pal et al. [9] proposed an algorithm for word-wise script identification from document containing *English, Devanagari* and *Telugu* text, based on conventional and water reservoir features. Chaudhury et al. [10] proposed a method for identification of Indian languages by combining Gabor filter based techniques and direction distance histogram classifier for *Hindi, English, Malayalam, Bengali, Telugu* and *Urdu*. Some analysis of the variability involved in the multi-script signature recognition problem as compared to the single-script scenario is discussed in [11,12].

Various classification algorithms are applied for different pattern recognition problems and the same fact also applies to the script recognition problem. Till date, for *Indic* script recognition purpose, different classifiers have been used such as *k*-Nearest Neighbours (*k*-NN) [13,14], Linear Discriminant Analysis (LDA) [15], Neural Networks (NN) [15,16], Support Vector Machine (SVM) [16,17], Tree based classifier [18,19], Simple Logistic [20] and MLP [21,22]. Though good results have already been achieved in this pattern recognition task but with a single classifier it is still hard to achieve acceptable accuracy. Studies expose that the fusion of multiple classifiers can be a viable solution to get better classification results as the error amassed by any single classifier is generally compensated using information from other classifiers. The reason for this is that different classifiers may offer complementary information about the patterns under consideration. Based on this fact, since long, a section of researchers has focused on devising different algorithms for combining classifiers in an intelligent way so that the combination can achieve better results than any of the individual classifier used for combining. The key idea is that instead of relying on a single decision maker, all the designs or their subsets are applied for the decision making by combining their individual beliefs in order to come up with a consensus decision. This fact motivates many researchers to apply the classifier combination methods to different pattern recognition problems. The popular methodologies for classifier combination include: Majority Voting [23,24], Subset-combining and re-ranking approach [25], Statistical model [26], Bayesian Belief Integration [27], Combination based on DS theory of evidence [27,28] and Neural Network combinator [29].

But till date, classifier combination approach for script recognition problem, either handwritten or printed, has not been tested much, though it has enormous potential. To bridge this research gap, this paper applies different classifier combination techniques in the field of *Indic* script recognition.

The main contribution of the present work is the comprehensive evaluation of the major classifier combination approaches which are either rule based or apply a secondary classifier for information fusion. The motivation is to improve the classification accuracy at the word-level handwritten script recognition by combining the results of the best performing classifier on three previously used feature sets. It is a multi-class classification problem and in the present case, 12 officially used *Indic*-scripts are considered which are: *Devanagari, Bangla, Odia, Gujarati, Gurumukhi, Tamil, Telugu, Kannada, Malayalam, Manipuri, Urdu* and *Roman*. Three different sets of feature vectors based on both shape and texture analysis have been estimated from each of the handwritten word images. Identification of the scripts in which the word images are written, is done with these feature values by feeding the same into different MLP classifiers. Soft-decisions provided by the individual classifiers are then combined using an array of classifier combination techniques. This kind of work is implemented for the first time assuming the number of *Indic* scripts undertaken and the range of combination techniques applied. The system developed for the script recognition task here, is a part of the general framework where different feature sets and classifier outputs can be modelled into a single system without much increase in the computation involved. Block diagram of the present work is shown in Figure 1.

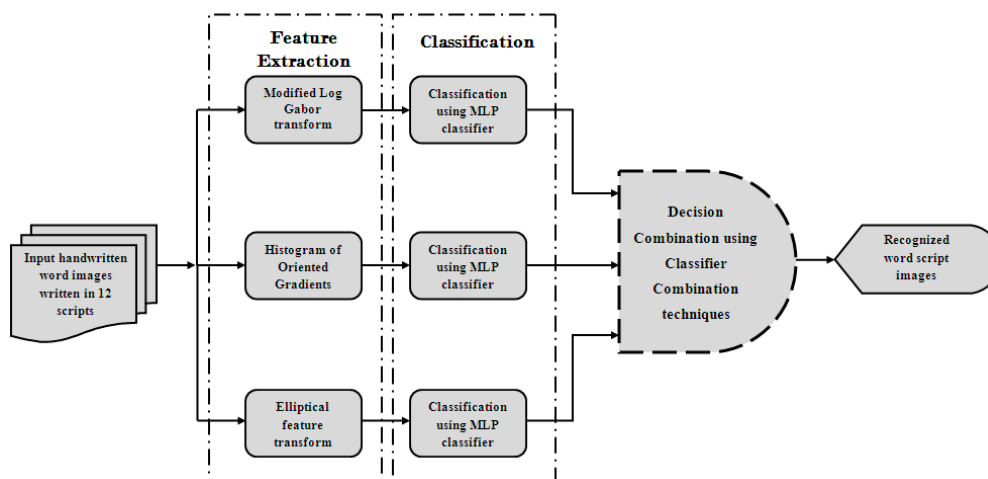


Figure 1. Schematic diagram of the proposed methodology.

2. Feature Extraction

In this paper, three popular feature extraction methodologies have been used for the combination *namely*, Elliptical Features [21], Histogram of Oriented Gradients (HOG) [30] and Modified log-Gabor filter transform [20]. The first feature set is applied to capture the overall structure present in the script word images whereas the rest two feature sets deal with the texture of the same. These features have already provided satisfactory results to this challenging task of handwritten script identification.

2.1. Elliptical Features

The word images are generally found to be elongated in nature which can better covered by an ellipse. That is why; elliptical features are extracted from the contour and the local regions of a word image so that it is easier to isolate a particular script. Two more important notations used in this subsection are: (a) Pixel ratio (P_r) and (b) Pixel count (P_c). P_r is defined as the ratio of the number of contour pixels (object) to the number of background pixels and the pixel count whereas P_c is defined as the number of contour pixels. The features are described in detail:

2.1.1. Maximum Inscribed Ellipse

The height and width of the bounding box are calculated for each word image. A representative ellipse is then inscribed (considering the orientation of the ellipse) inside this bounding box having

major and minor axes equal to the width and the height of the bounding box and the centre of an ellipse is also the centre of the corresponding bounding box. This ellipse divides the word image into eight regions $R_i, i = 1, 2 \dots, 8$. The bounding box along with the inscribed ellipse for a handwritten Bangla word image are shown in Figure 2b. Taking the values of P_r from these eight regions, as shown in Figure 2a, eight features (F1–F8) for each handwritten word image are estimated. Now, another type of feature, P_c along N ($N = 8$ for the present work) lines parallel to major/minor axis of the representative ellipse are computed. The mean and standard deviation of the values of P_c along major/minor axis are taken as four additional features (F9–F12).

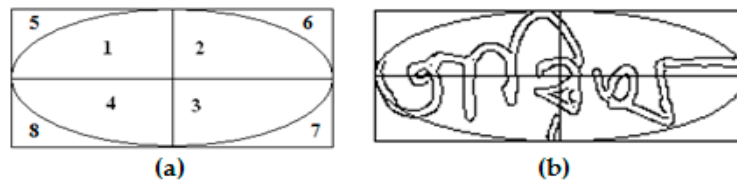


Figure 2. Illustration of fitting (a) an imaginary ellipse inside the minimum boundary box which divides a Bangla handwritten word image in 8 regions as shown in (b).

2.1.2. Sectional Inscribed Ellipse

Each of the word images surrounded by the minimum bounding box is again divided into four equal rectangles and a representative ellipse is fit into each of these rectangles using the same procedure as described in the previous subsection. As a result, every ellipse produces eight regions inside its rectangular area namely, R_{ij} where $1 \leq i \leq 4$ and $1 \leq j \leq 8$ which makes $8 \times 4 = 32$ regions in total. A total of 32 feature values (F13–F44) using the P_r values is computed from the 32 ellipses in similar fashion.

2.1.3. Concentric Ellipses

These feature values are computed by taking the entire topology of the word image. A primary ellipse is made circumscribing the word image with centre taken to be the midpoint of its minimum bounding box. The values of the major and minor axes of the ellipse are taken into consideration. After fitting the primary ellipse, three concentric ellipses are drawn inside the primary ellipse having the same centre point as the primary ellipse and major and minor axes equal to 1/4th, 2/4th and 3/4th of major and minor axes of the primary ellipse respectively. These four ellipses divide each of the word images into four regions- R_{e1}, R_{e2}, R_{e3} and R_{e4} . The partitioning of the four regions on a sample handwritten *Devanagari* word image is shown in Figure 3. From the four regions, four features values (F45–F48) considering the P_r 's and four feature values (F49–F52) considering the P_c 's of the regions R_{e1}, R_{e2}, R_{e3} and R_{e4} are estimated. The remaining six features (i.e., F53–F58) are taken as the corresponding differences of the P_r 's and P_c 's between the regions R_{e1} and R_{e2}, R_{e2} and R_{e3}, R_{e3} and R_{e4} respectively. The elliptical features (F1–F58) are suitably normalized by the height and width of the corresponding word image.

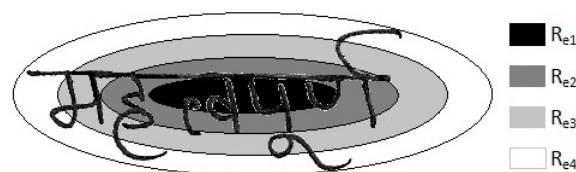


Figure 3. Figure showing the elliptical partition of four regions on a sample handwritten *Devanagari* word image.

2.2. Histogram of Oriented Gradients (HOG)

HOG descriptor [31] counts occurrences of gradient orientation in localized portions of an image which was first proposed for pedestrian detection in steady images. The essential thought behind the HOG descriptor is that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. At first, the values of the magnitude and direction of all the pixels for each of the word images are calculated. Next, each pixel is pigeonholed in certain category according to its direction which is known as orientation bins. Then, the word image is divided into n (here $n = 10$) connected regions, called cells and for each cell, a histogram of gradient directions or edge orientations is computed for the pixels within the cell. The combination of these histograms then represents the descriptor. Since the number of orientation bins is taken as 8 for the present work, an 80-D (i.e., 10×8) feature vector has been extracted using HOG descriptor [30]. The magnitude and direction of each pixel of a sample handwritten *Telugu* word image are also shown in Figure 4.

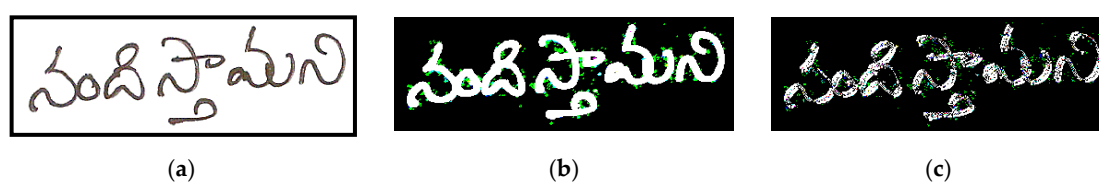


Figure 4. Illustration of: (a) handwritten *Telugu* word image, (b) its magnitude part and (c) its direction part.

2.3. Modified Log-Gabor Filter Transform (MLG Transform)

Modified log-Gabor filter transform-based features, proposed in Reference [20], had performed well in the script classification task and therefore are also chosen as one of the feature descriptors of our proposed methodology in order to identify the script of the word images. In order to preserve the spatial information, a Windowed Fourier Transform (WFT) is considered in the present work. WFT involves multiplication of the image by the window function and the resultant output is followed by applying the Fourier transform. WFT is basically a convolution of the image with the low-pass filter. Since for texture analysis, both spatial and frequency information are preferred, the present work tries to achieve a good trade-off between these two. Gabor transforms use a Gaussian function as the optimally concentrated function in the spatial as well as in the frequency domain [32]. Due to the convolution theorem, the filter interpretation of the Gabor transform allows the efficient computation of the Gabor coefficients by multiplication of the Fourier transformed image with the Fourier transform of the Gabor filter. The inverse Fourier transform is then applied on the resultant vector to get the output filtered images.

The images, after low pass filtering, are passed as input to a function that computes Gabor energy feature from them. The input image is then passed to a function to yield a Gabor array which is the array equivalent of the image after Gabor filtering. The function displays the image equivalent of the magnitude and the real part of the Gabor array pixels.

For the present work, both energy and entropy features [33] based on Modified log-Gabor filter transform have been extracted for 5 scales (1, 2, 3, 4 and 5) and 6 orientations (0° , 30° , 60° , 90° , 120° and 150°) to capture complementary information found in different script word images. Here, each filter is convolved with the input image to obtain 60 different representations (response matrices) for a given input image. Figure 5 shows output images formed after the application of Modified log-Gabor filter transform for a sample handwritten *Bangla* word image.

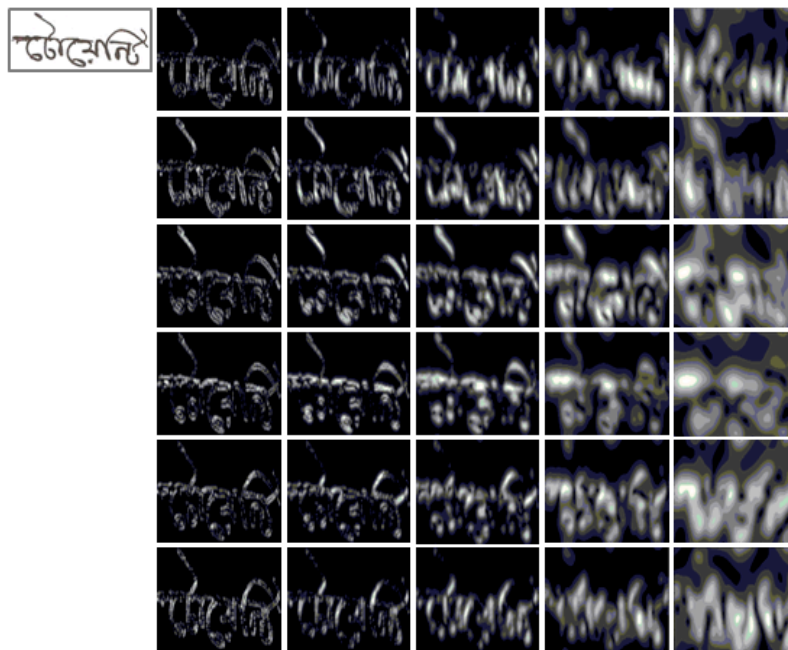


Figure 5. Output word images of Modified log-Gabor filter transform on a sample handwritten *Bangla* word image (shown on left-side) for 5 different scales and 6 different orientations (the first row shows the output for $n_o = 0^0$ and five scales, the second row shows the output for $n_o = 30^0$ and five scales and so on).

3. Classifier Combination

Classifier combination tries to improve on the task of pattern recognition performance through mathematical models. The outputs of classifiers can be represented as vectors of numbers where the dimension of vectors is equal to the number of classes. As a result, the combination problem can be defined as a problem of finding the combination function accepting N-dimensional score vectors from M classifiers and outputting N final classification scores (see Figure 6), where the function tries to minimize the misclassification cost.

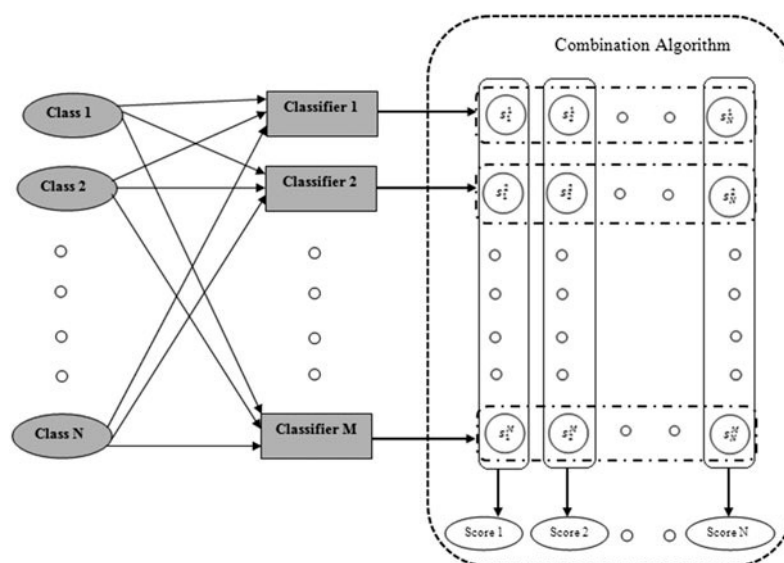


Figure 6. Classifier combination takes a set of s_i^j score for class i by classifier j and produces combination scores S_i for each class i [34].

The field of classifier combination can be grouped into different categories [35] based on the stage at which the process is applied, type of information (classifier output) being fused and the number and type of classifiers being combined.

Based on the operating level of the classifiers, classifier combination can be done at the feature level. Multiple features can be joined to provide a new feature set which provide more information about the classes. But with the increase in dimensionality of the data, training becomes expensive.

The classifier outputs after the extraction of the individual feature sets can be combined to provide better insights at the decision level. Decision level combination techniques are popular as it cannot need any understanding of the ideas behind the feature generation and classification algorithms.

Feature level combination is performed by concatenating the feature sets in all possible combinations and passing it through the base classifier, MLP in this case. Apart from that, all the other combination processes worked out operate at the decision level.

Classifier combination can also be classified by the outputs of the classifiers used in the combination. Three types of classifier outputs are usually considered [36]:

- Type I (Abstract level): This is the lowest level in a sense that the classifier provides the least amount of information on this level. Classifier output is a single class label informing the decision of the classifier.
- Type II (Rank level): Classifier output on the rank level is an ordered sequence of candidate classes, the so-called n-best list. The candidate classes are ordered from the most likely class at the front and the least likely class index featuring at the last of the list. There are no confidence scores attached to the class labels on rank level and the relative positioning provides the required information.
- Type III (Measurement level): In addition to the ordered n-best lists of candidate classes on the rank level, classifier output on the measurement level has confidence values assigned to each entry of the n-best list. These confidences, or scores, are generally real numbers generated using the internal algorithm for the classifier. This soft-decision information at the measurement level thus provides more information than the other levels.

In this paper, Type II (rank level) and Type III (measurement level) combination procedures are worked out because they allow the inculcation of a greater degree of soft-decision information from the classifiers and find use in most practical applications.

The focus of this paper is to explore the classifier combination techniques on a fixed set of classifiers. The purpose of the combination algorithm is to learn the behaviour of these classifiers and produce an efficient combination function based on the classifier outputs. Hence, we use non-ensemble classifier combinations which try to combine heterogeneous classifiers complementing each other. The advantage of complementary classifiers is that each classifier can concentrate on its own small sub problem and together the single larger problem is better understood and solved. The heterogeneous classifiers, here, are generated by training the same classifier with different feature sets and tuning them to optimal values of their parameters. This procedure does away with the need for normalization of the confidence scores provided by different classifiers which do not tend to follow a common standard and depend of the algorithm. For example, in the MLP classifier used here, the last layer has each node containing a final score for one class. These scores can then be used for the rank level and decision level combination along with the maximum being chosen for the individual classifier decision.

In the next sub-section, the set of major classification algorithms evaluated in this paper are categorized into two approaches based on how the combination process is implemented. In the first approach, rule based combination practices are demonstrated that apply a given function to combine the classifier confidences into a single set of output scores. The second approach employs another classifier, called the 'secondary' classifier that operates on the outputs of the base classifier and automatically account for the strengths of the participants. The classification algorithm is trained on these confidence values with output classes same as the original pattern recognition problem.

Essentially, both the approaches apply a function on the confidence score inputs, where the rule based functions are simpler operations like sum rule, max rule, etc. and classifiers like k-NN and MLP apply more complicated functions.

3.1. Rule Based Combination Techniques

Rules are applied on the abstract level, rank level and measurement level outputs from the classifiers to obtain a final set of confidence scores that can take into account the insights provided by the previous stage of classification. Elementary combination approaches like majority voting, Borda count, sum rule, product rule and the max rule come under this approach of classifier combination. DS theory of evidence is a relatively complex technique that is adopted for this purpose, utilising the rule of combination for information sources with the same frame of discernment.

3.1.1. Majority Voting

A straightforward voting technique is majority voting operating at the abstract level. It considers only the decision class provided by each classifier and chooses the most frequent class label among this set. In order to reduce the number of ties, the number of classifiers used for voting is usually odd.

3.1.2. Borda Count

Borda count is a voting technique on rank level [37]. For every class, Borda count adds the ranks in the n-best lists of each classifier so that for every output class the ranks across the classifier outputs get accumulated. The class with the most likely class label, contributes the highest rank number and the last entry has the lowest rank number. The final output label for a given test pattern X is the class with highest overall rank sum. In mathematical terms, this reads as follows: Let N be the number of classifiers and r_i^j the rank of class i in the n-best list of the j -th classifier. The overall rank r_i of class i is thus given by

$$r_i = \sum_{j=1}^N r_i^j \quad (1)$$

The test pattern X is assigned the class i with the maximum overall rank count r_i . Borda count is very simple to compute and requires no training. There is also a trainable variant that associates weights to the ranks of individual classifiers. The overall rank count for class i is then computed as given below

$$r_i = \sum_{j=1}^N w_j r_i^j \quad (2)$$

The weights can be the performance of each individual classifier measured on a training or validation set.

3.1.3. Elementary Combination Approaches on Measurement Level

Elementary combination schemes on measurement level apply simple rules for combination, such as sum rule, product rule and max rule. Sum rule simply adds the score provided by each classifier from a set of classifier for every class and assigns the class label with the maximum score to the given input pattern. Similarly, product rule multiplies the score for every class and then outputs the class with the maximum score. The max rule predicts the output by the selecting the class corresponding to the maximum confidence value among all the participating classifiers' output scores.

Interesting theoretical results, including error estimations, have been derived for these simple combination schemes. Kittler et al. showed that sum rule is less sensitive to noise than other rules [38]. Despite their simplicity, simple combination schemes have resulted in high recognition rates and shown comparable results to the more complex procedures.

3.1.4. Dempster-Shafer Theory of Evidence

The DS framework [39] is based on the view whereby propositions are represented as subsets of a given set W , referred to as a frame of discernment. Evidence can be associated to each proposition (subset) to express the uncertainty (belief) that has been observed or discerned. Evidence is usually computed based on a density function m called Basic Probability Assignment (BPA) and $m(p)$ represents the belief exactly committed to the proposition p .

DS theory has an operation called *Dempster's rule of combination* that aggregates two (or more) bodies of evidence defined within the same frame of discernment into one body of evidence. Let m_1 and m_2 be two BPAs defined in W . The new body of evidence is defined by the $BPA_{m_1,2}$ as:

$$m_{1,2}(A) = \begin{cases} 0 & \text{if } A = \emptyset \\ \frac{1}{1-K} \sum_{B \cap C = A} m_1(B)m_2(C) & \text{if } A \neq \emptyset \end{cases} \quad (3)$$

where, $K = \sum_{B \cap C = \emptyset} m_1(B)m_2(C)$ and A is the intersection of subsets B and C .

In other words, the Dempster's combination rule computes a measure of agreement between two bodies of evidence concerning various propositions determined from a common frame of discernment. The rule focuses only on those propositions that both bodies of evidence support.

The denominator is a normalization factor that ensures that m is a BPA, called the conflict. The Yagar's modification of the DS theory [40] has been implemented in the paper with the normalizing factor as 1. This reduces some of the issues regarding the conflict factor.

Earlier, DS theory based combination has been applied on different fields like handwritten digit recognition [41], skin detection [42], 3D palm print recognition [43] among other pattern recognition domains.

3.2. Secondary Classifier Based Combination Techniques

The confidence values provided by the classifiers act as the feature set for the secondary classifier which acts on the second stage of the framework. With the training from the classifier scores, it learns to predict the outcome for a set of new confidence scores from the same set of classifiers. The advantage of using such a generic combinator is that it can learn the combination algorithm and can automatically account for the strengths and score ranges of the individual classifiers. For example, Dar-Shyang Lee [29] used a neural network to operate on the outputs of the individual classifiers and to produce the combined matching score. Apart from the neural network, other classifiers like k -NN, SVM and Random Forest have been fitted and tested in this paper.

4. Results and Interpretation

4.1. Preparation of Database

At present, no standard benchmark database of handwritten *Indic* scripts is freely available in the public domain. Hence, we have created our own database of handwritten documents in the laboratory. The document pages for the database were collected from different sources on request. Participants of this data collection drive were asked to write few lines on A-4 size pages. No other restrictions were imposed regarding the content of the textual materials. The documents were written in 12 official scripts of India. The document pages are digitized at 300 dpi resolution and stored as grey tone images. The scanned images may contain noisy pixels which are removed by applying Gaussian filter [33]. The text words are automatically extracted from the handwritten documents by using a page-to-word segmentation algorithm described in [44]. A sample snapshot of word images written in 12 different scripts is shown in Figure 7. Finally, a total of 7200 handwritten word images are prepared, with exactly 600 text words per script.

বিক্রম	বিক্রম	বিক্রম	বিক্রম	বিক্রম
<i>Bangla</i>				
लेकिन	प्रत्युष	यात्रियों	विकेट	नेतृत्व
<i>Devanagari</i>				
જવાની	સુલાકાતો	સવારે	અનુક્રમ	તહેવાર
<i>Gujarati</i>				
ਮੰਦੇਸ਼	ਗਮਿੰਦਰ	ਪਾਇਓ	ਚਮੜੇਰ	ਖਮਪਰ
<i>Gurumukhi</i>				
ಮುಖ್ಯವಾಗಿ	ಕರ್ತವ್ಯ	ಲಕ್ಷ್ಯಂಕರ	ಉದಾಹರಣೆ	ಮುಖ್ಯ
<i>Kannada</i>				
നിറഞ്ഞ	കുറഞ്ഞ	ഇതിൽ	ബന്ധപ്പെട്ട	നിരക്കുകളുടേ
<i>Malayalam</i>				
ꯀꯪ꯰ꯪ꯰	ꯪ꯰ꯪ꯰ꯪ	ꯪ꯰ꯪ꯰ꯪ	ꯪ꯰ꯪ꯰ꯪ	ꯪ꯰ꯪ꯰ꯪ
<i>Manipuri</i>				
ଧରଣୀ	ଭୂମି	ଭୂମି	ଦର୍ଶନ	ମନୁଷ୍ୟ
<i>Oriya</i>				
வரணம்	கருதல்	கும்பகந்த	பெரியமை	கேவலம்
<i>Tamil</i>				
వారికే	యదినను	సంబంధం	ఉత్తరము	చైతన్యం
<i>Telugu</i>				
مگر حیدرآباد	کھیلا	کھیلا	کھیلا	کھیلا
<i>Urdu</i>				
substances	dominating	which	chief	third
<i>Roman</i>				

Figure 7. Sample word images written in 12 different Indian scripts.

Our developed database has been named as *CMATERdb8.4.1*, where *CMATER* stands for ‘Centre for Microprocessor Applications for Training Education and Research,’ a research laboratory at Computer Science and Engineering Department of Jadavpur University, India, where the current database is prepared. Here, *db* symbolizes database, the numeric value 8 represents handwritten multi-script *Indic* image database and the value 4 indicates word-level. In the present work, the first version of *CMATERdb8.4* has been released as *CMATERdb8.4.1*. The database is made freely available at <https://code.google.com/p/cmaterdb/>.

4.2. Performance Analysis

The classifier combination approaches, described above, are applied on a dataset of 7200 words divided into 12 classes with equal number of instances in each of them. 12 classes refer to the 12 *Indic* scripts that have been studied before and for which the MLP classifier results can be obtained with high accuracy. The classes numbered from **A** to **L** are *Devanagari*, *Bangla*, *Oriya*, *Gujarati*, *Gurumukhi*, *Tamil*, *Telugu*, *Kannada*, *Malayalam*, *Manipuri*, *Urdu* and *Roman* in that particular order.

First, the confusion matrix that is obtained from the MLP based classifier on the dataset by using MLG feature along with the overall accuracy is presented. Then, the result generated by the same classifier on the HOG and Elliptical feature sets applied on the same dataset is also presented. Results have been cross-validated for the classifier parameter values to obtain the optimal results for the dataset and the values are provided in the result section.

The MLG feature set consisting of 60 feature values for every input image is fed into the MLP classifier with 30 hidden layer neurons and a learning rate of 0.8. Here, 500 iterations are allowed with an error tolerance of 0.1. The overall accuracy obtained is 91.42% and the confusion matrix generated in this case is given in Table 1. The **R** column in the table refers to the rejection of the input by the recognition module but the class confidences that are associated with them get accounted for during the combination process.

The HOG feature set, consisting of 80 feature values for every input data, is fed into the MLP classifier with 40 hidden layer neurons and a learning rate of 0.8. Same error tolerance and the number of iterations, as applied in case of MLG features, are allowed here. A maximum recognition accuracy of 78.04% has been noted. The confusion matrix is shown in Table 2.

The Elliptical feature set containing 58 feature values derived from each image data forms the training set for the MLP classifier with 30 hidden neurons with a learning rate of 0.7. The error tolerance and number of iterations remain the same as the previous cases. An accuracy of 79.2% is achieved and represented in the confusion matrix given in Table 3.

Table 1. Classification results for HOG feature set with MLP Classifier.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L	R
A	345	9	6	22	13	21	64	42	27	0	44	7	27
B	27	548	0	7	9	0	1	0	1	0	7	0	0
C	0	0	557	0	6	13	1	19	2	1	0	1	38
D	38	4	0	516	3	3	4	0	9	0	20	3	10
E	10	6	1	12	449	26	5	2	0	0	13	76	30
F	30	0	23	3	46	417	33	36	6	1	4	1	27
G	27	2	15	10	12	16	446	34	12	1	24	1	10
H	10	0	27	17	16	41	8	420	28	11	14	8	38
I	38	2	4	16	0	10	34	33	455	0	8	0	0
J	0	0	17	0	7	0	0	16	0	553	1	6	38
K	38	6	5	35	22	14	42	31	0	2	404	1	2
L	2	2	14	6	15	24	1	9	0	13	5	509	0

Table 2. Classification results for MLG feature set with MLP Classifier.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L	R
A	528	0	2	13	1	1	19	9	5	0	12	10	0
B	0	576	0	6	0	0	0	0	0	0	3	15	1
C	1	0	596	0	0	0	1	1	1	0	0	0	2
D	2	9	0	574	0	0	0	0	1	0	0	14	0
E	0	0	0	0	592	6	0	1	0	0	0	1	0
F	0	0	2	0	16	553	0	20	0	9	0	0	4
G	4	0	9	3	0	1	528	15	26	0	10	4	7
H	7	0	5	0	5	30	8	512	16	1	8	8	12
I	12	0	7	1	0	0	12	2	560	0	4	2	0
J	0	0	0	0	3	4	0	5	0	588	0	0	19
K	19	3	1	7	2	0	24	2	4	0	527	11	3
L	3	2	25	29	24	9	4	21	18	4	13	448	0

Table 3. Classification results for Elliptical feature set with MLP Classifier.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L	R
A	355	29	49	48	0	25	3	4	42	10	6	29	2
B	2	550	0	7	0	1	32	0	0	0	0	8	27
C	27	0	479	8	1	19	2	11	31	1	8	13	32
D	32	9	0	514	0	13	23	0	3	2	4	0	66
E	66	1	2	1	441	42	4	7	10	20	4	2	96
F	96	3	6	15	6	397	16	4	19	12	14	12	55
G	55	13	7	54	6	17	402	1	3	19	22	1	25
H	25	0	2	3	0	26	0	491	28	10	4	11	7
I	7	0	23	3	33	8	7	3	493	10	4	9	0
J	0	0	1	0	16	5	2	2	9	553	6	6	2
K	2	0	16	7	1	7	12	0	2	7	546	0	8
L	8	22	1	0	6	6	20	13	6	9	1	508	0

Now, the confidence values provided to the classes for every input data by the classifiers on the three sets of features form the input for the classifier combination procedures. The confusion matrix resulting from the Majority voting procedure is presented in Table 4. An overall accuracy of 95.6% is achieved on this dataset containing 7200 samples divided equally among the 12 script classes. It is seen that *Devanagari* script has got the least accuracy and gets confused with *Telugu* whereas high accuracies are shown for *Manipuri* and *Odia* and *Bangla*.

Borda count algorithm gives an accuracy of 93.5% which is an increase of 2.1% over the best performing individual classifier. It provides the highest recognition rate for *Devanagari* among all the combination schemes and good accuracies for other popular scripts like *Bangla* and *Odia* and hence can be the preferred choice for wide usage. The trainable version of the algorithm with weights based on overall accuracy of the classifiers improves the results further. The increase is 2.9% with satisfactory results for scripts like *Telugu*, *Kannada* and *Urdu*. The accuracy for the *Gurumukhi* script remains low irrespective of the weights. The results are presented in Tables 5 and 6.

The simple rules at the measurement level to combine the decisions provide good results in the present work. The sum rule attains an accuracy of 97.76% with almost close to perfect recognition for *Urdu*, *Gurumukhi* and *Roman*. The product rule and max rule have accuracies of 95.73% and 94.60% respectively. Highest accuracy is found for *Odia* script whereas product rule suffers in case of *Gurumukhi* and max rule in case of *Devanagari*. The results for the elementary rules of combination are tabulated in Tables 7–9.

Sum rule outperforms all other rule based combination approaches in this work and testifies the results presented by Kittler et al. mentioned in [38] by being less prone to noise and unclean data. The DS theory results combine the results, two at a time and then all three together. The class-wise performance based BPA, which outperforms the global performance based BPA, has been implemented for the multi-classifier combination using the DS theory [45]. The rule applied for this process is quasi-associative and hence the results of combining two sources cannot be combined with the third. The rule has to be extended to include all the three sources together. Results for the combination of the classifier results on HOG and Elliptical features, MLG and Elliptical features, and, HOG and MLG features are presented in Tables 10, 11 and 12 respectively. The combination result including all the three sources of information is given in Table 13.

There is no improvement shown by the combination of the results from MLG and HOG feature sets. But when the Elliptical feature set is involved in the combination process there is much improvement over the participating classifiers. Overall accuracies of 91.2% and 97.04% are achieved by combining sources having 78.1% and 79.4% accuracies and 91.4% and 79.4% accuracies respectively. So, improvements of 6% and 10% are found by applying the DS theory of evidence. Combining all three, an accuracy of 95.64%, more than 4% over the better performing classifier is seen. In both the schemes all the script classes have accuracies over 90% and with almost 100% accuracy for certain

scripts like *Manipuri*, *Gujarati* and *Urdu*, thus proving to be the model to be used where these scripts are widely used.

In order to understand why the results from the Elliptical feature set combine so well with the two other feature sets, correlation analysis is performed on the confidence score outputs. Spearman rank correlation is done on the rank level information provided by the classifiers to arrive at mean values for the measure of the correlation. HOG and MLG show an index of 0.619 which is almost the double of the scores obtained by comparing the Elliptical features with these two. With values of 0.32 and 0.27, the low correlation index is an indication of better possibilities for the combination processes. Thus, complementary information is provided by the output of Elliptical feature set which helps in the improvement the overall combined accuracy.

Secondary classifiers are applied to learn the patterns from the primary classifier outputs and develop a way to combine them. The confidence scores from the three sources are concatenated to form a larger training set with its correct label. This set is the new feature set which undergoes classification using well-known algorithms. Classifiers like *k*-NN, Logistic Regression, MLP and Random Forest are applied to report final results which are tabulated in Tables 14–17 respectively. The results are reported after 3-fold cross validation and tuning of the parameters involved. This process is computationally costly and takes a processing step along with much higher complexity but is compensated by the high accuracy results that are obtained. 3-NN provides an accuracy of 98.30%, Random Forest classified 98.33% of the 7200 samples correctly and Logistic Regression attained 98.48% accuracy. Using MLP again as the secondary classifier, 98.36% accuracy is obtained. *Devanagari* is the most confused script in all the cases but still has accuracy over 95%. The other scripts are predicted to almost certainty.

Table 4. Classification results after combination using Majority voting procedure.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L
A	534	2	3	10	3	2	16	7	13	0	5	5
B	1	590	0	3	0	0	0	0	0	0	0	6
C	0	0	597	0	0	2	0	0	1	0	0	0
D	2	3	0	590	0	0	0	0	0	0	2	3
E	0	1	0	0	591	2	0	0	0	1	0	5
F	12	0	5	0	13	561	1	7	1	0	0	0
G	2	0	6	4	5	1	554	14	7	3	4	0
H	4	0	4	3	1	6	0	567	7	2	5	1
I	9	1	1	2	0	1	7	2	572	0	5	0
J	0	0	1	0	2	0	0	3	0	594	0	0
K	4	0	2	5	4	1	10	4	1	0	567	2
L	0	2	10	3	6	2	1	1	1	5	3	566

Table 5. Classification results after combination using Borda count procedure without weight.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L
A	567	0	5	7	0	0	5	4	7	0	3	2
B	16	580	0	4	0	0	0	0	0	0	0	0
C	1	0	586	0	0	5	0	4	3	0	0	1
D	25	1	0	572	0	0	0	0	0	0	1	1
E	6	0	0	0	466	108	0	11	1	0	0	8
F	16	0	2	0	7	571	0	1	1	1	0	1
G	25	0	4	2	0	2	548	3	1	0	15	0
H	30	0	5	0	0	21	0	533	6	0	1	4
I	39	0	2	1	0	2	6	6	540	0	4	0
J	0	0	2	0	2	0	0	7	0	589	0	0
K	5	0	0	3	0	0	12	0	1	0	579	0
L	4	0	10	2	2	4	0	1	3	3	3	568

Table 6. Classification results after combination using Borda count procedure using weight.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L
A	563	0	4	7	0	0	8	6	7	0	3	2
B	14	582	0	4	0	0	0	0	0	0	0	0
C	0	0	586	0	0	5	0	5	3	0	0	1
D	15	2	0	580	0	0	0	0	0	0	2	1
E	5	0	0	0	466	102	0	17	2	0	0	8
F	13	0	0	0	6	576	0	2	1	1	0	1
G	14	0	3	2	0	1	558	3	1	0	18	0
H	22	0	5	0	0	14	0	546	6	0	1	6
I	30	0	2	1	0	1	6	5	551	0	4	0
J	0	0	2	0	2	0	0	6	0	590	0	0
K	2	0	0	2	0	0	9	0	0	0	587	0
L	4	0	10	1	2	4	0	1	2	3	3	570

Table 7. Classification result after combination using Sum rule.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L
A	549	1	4	6	0	1	13	10	11	0	3	2
B	3	589	0	2	0	0	0	0	0	0	1	5
C	0	0	597	0	0	1	0	1	1	0	0	0
D	1	3	0	593	0	0	0	0	0	0	0	3
E	0	0	0	0	595	2	0	0	0	0	0	3
F	6	0	2	0	8	576	0	7	0	0	0	1
G	3	0	5	3	1	0	568	10	3	2	5	0
H	2	0	2	0	0	5	0	582	4	1	3	1
I	16	0	1	1	0	1	6	3	569	0	3	0
J	0	0	0	0	0	0	0	1	0	599	0	0
K	0	0	0	2	2	0	5	3	0	0	588	0
L	1	0	6	1	3	0	0	0	1	3	2	583

Table 8. Classification result after combination using Product rule.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L
A	566	0	4	6	0	0	6	6	9	0	1	2
B	7	584	0	8	0	0	0	0	1	0	0	0
C	0	0	597	0	0	2	0	0	1	0	0	0
D	5	2	0	591	0	0	0	0	0	0	1	1
E	4	0	0	0	467	97	0	10	2	0	7	13
F	8	0	0	0	5	582	0	3	0	1	0	1
G	5	0	2	2	0	1	578	3	1	0	8	0
H	12	0	4	0	0	15	0	562	4	0	0	3
I	31	0	0	0	0	1	8	2	556	0	2	0
J	0	0	1	0	1	0	0	4	0	594	0	0
K	2	0	0	1	0	0	4	0	0	0	593	0
L	2	0	2	0	0	1	0	1	0	2	3	589

Table 9. Classification result after combination using Max rule.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L
A	512	6	4	13	4	2	17	13	17	0	8	4
B	1	587	0	3	0	0	2	0	0	0	0	7
C	0	0	599	0	1	0	0	0	0	0	0	0
D	2	5	0	580	0	0	2	0	3	0	3	5
E	1	1	0	4	566	3	0	1	0	1	0	23
F	7	0	6	0	10	556	2	12	2	1	1	3
G	3	1	6	5	4	1	553	8	7	4	8	0
H	3	0	4	4	1	5	0	566	11	2	3	1
I	8	1	1	1	2	1	6	5	569	0	5	1
J	0	0	1	0	1	0	0	4	0	594	0	0
K	3	1	4	5	6	4	5	6	1	0	564	1
L	0	3	3	9	5	2	2	2	2	5	2	565

Table 10. Classification results after combination using DS theory for HOG and Elliptical features.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L
A	512	6	4	13	4	2	17	13	17	0	8	4
B	1	587	0	3	0	0	2	0	0	0	0	7
C	0	0	599	0	1	0	0	0	0	0	0	0
D	2	5	0	580	0	0	2	0	3	0	3	5
E	1	1	0	4	566	3	0	1	0	1	0	23
F	7	0	6	0	10	556	2	12	2	1	1	3
G	3	1	6	5	4	1	553	8	7	4	8	0
H	3	0	4	4	1	5	0	566	11	2	3	1
I	8	1	1	1	2	1	6	5	569	0	5	1
J	0	0	1	0	1	0	0	4	0	594	0	0
K	3	1	4	5	6	4	5	6	1	0	564	1
L	0	3	3	9	5	2	2	2	2	5	2	565

Table 11. Classification results after combination using DS theory for Elliptical and MLG features.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L
A	564	0	3	8	0	1	7	4	8	0	0	5
B	0	593	0	1	0	0	0	0	0	0	0	6
C	0	0	600	0	0	0	0	0	0	0	0	0
D	0	7	0	590	0	0	1	0	0	0	0	2
E	1	0	0	0	589	8	0	1	0	1	0	0
F	0	0	0	0	5	580	1	7	0	4	1	2
G	5	0	4	3	0	0	563	3	10	3	8	1
H	3	0	1	0	1	16	0	567	7	2	1	2
I	6	0	1	0	2	0	2	1	583	2	1	2
J	0	0	0	0	0	0	0	0	0	600	0	0
K	1	0	1	0	0	0	5	0	1	0	592	0
L	3	6	3	3	7	0	4	2	2	2	2	566

Table 12. Classification results after combination using DS theory for HOG and MLG features.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L
A	511	2	4	12	5	1	19	19	12	0	8	7
B	6	573	0	5	2	0	0	0	0	0	5	9
C	1	0	590	0	1	1	0	5	2	0	0	0
D	4	4	0	580	0	0	1	0	2	0	4	5
E	3	1	1	6	540	6	0	1	0	0	3	39
F	7	0	5	0	18	549	1	18	0	2	0	0
G	7	1	7	5	2	0	530	21	11	1	13	2
H	10	0	10	6	8	13	4	524	11	5	7	2
I	26	1	2	4	0	0	17	11	535	0	4	0
J	0	0	3	0	5	0	0	9	0	580	0	3
K	22	3	4	11	3	1	15	11	3	2	522	3
L	3	1	19	8	12	4	1	2	2	6	3	539

Table 13. Classification results after combination using DS theory for Elliptical, HOG and MLG features.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L
A	548	1	5	7	3	1	12	5	12	0	2	4
B	4	584	0	5	0	0	0	0	0	0	1	6
C	0	0	598	0	0	0	0	1	1	0	0	0
D	5	2	0	592	0	0	0	0	0	0	0	1
E	0	1	0	5	548	22	0	3	1	2	3	15
F	10	1	2	0	5	572	1	6	0	1	0	2
G	10	0	3	3	3	0	556	9	7	1	8	0
H	8	0	4	0	2	8	0	568	4	1	4	1
I	17	1	1	2	0	0	12	3	561	0	3	0
J	0	0	0	0	1	0	0	2	0	597	0	0
K	6	0	1	2	0	2	7	0	0	0	582	0
L	2	1	5	4	2	0	2	0	0	3	1	580

Table 14. Classification results using 3-NN secondary classifier.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L
A	573	1	3	5	0	0	8	5	3	0	1	1
B	1	597	0	2	0	0	0	0	0	0	0	0
C	1	0	598	0	0	0	0	0	1	0	0	0
D	1	2	0	595	0	0	0	0	0	0	0	2
E	0	0	0	0	599	1	0	0	0	0	0	0
F	0	0	0	0	4	590	0	5	0	0	0	1
G	7	1	5	3	0	0	571	1	5	1	6	0
H	2	0	0	0	0	9	1	585	1	0	2	0
I	11	0	2	0	0	0	0	0	586	0	1	0
J	0	0	0	0	0	0	0	2	0	598	0	0
K	1	0	0	0	1	0	4	0	0	0	594	0
L	1	0	3	0	1	1	0	1	0	1	0	592

Table 15. Classification results using Logistic Regression secondary classifier.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L
A	583	0	1	2	0	0	5	4	2	0	1	2
B	0	598	0	2	0	0	0	0	0	0	0	0
C	1	0	598	0	0	0	0	0	1	0	0	0
D	1	2	0	594	0	0	0	1	0	0	0	2
E	1	0	0	0	595	3	0	0	0	0	0	1
F	0	0	0	0	4	589	0	5	1	0	0	1
G	4	0	3	2	0	0	578	2	3	0	8	0
H	3	0	0	0	0	6	1	586	1	0	1	2
I	6	0	1	0	0	0	2	3	587	0	1	0
J	0	0	0	0	0	2	0	1	0	596	0	1
K	1	0	0	0	0	0	5	0	0	0	594	0
L	1	0	1	0	2	0	2	1	0	0	0	593

Table 16. Classification results using MLP secondary classifier.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L
A	575	1	2	4	0	0	4	4	4	0	2	4
B	1	597	0	2	0	0	0	0	0	0	0	0
C	1	0	598	0	0	0	0	0	1	0	0	0
D	3	2	0	594	0	0	0	1	0	0	0	0
E	0	0	0	0	599	1	0	0	0	0	0	0
F	0	0	0	0	2	594	0	3	0	0	0	1
G	7	1	2	1	3	0	577	0	2	1	6	0
H	4	0	0	0	1	7	1	583	1	1	0	2
I	7	0	2	1	0	0	3	0	586	0	1	0
J	0	0	1	0	0	0	0	0	0	599	0	0
K	0	0	0	0	0	0	7	0	0	0	593	0
L	1	0	0	0	4	0	1	0	0	0	0	594

Table 17. Classification results using Random Forest secondary classifier.

Class \ Class	A	B	C	D	E	F	G	H	I	J	K	L
A	581	0	3	2	0	0	8	1	4	0	0	1
B	0	597	0	2	0	0	1	0	0	0	0	0
C	1	0	595	0	0	0	2	0	1	0	0	1
D	2	2	0	593	0	0	0	0	0	0	0	3
E	0	0	0	0	598	2	0	0	0	0	0	0
F	0	0	0	0	7	585	0	6	0	1	0	1
G	3	0	4	1	0	0	585	1	2	0	4	0
H	4	0	2	0	0	8	0	582	1	0	1	2
I	9	0	1	0	0	0	3	2	584	0	1	0
J	0	0	0	0	0	2	0	2	0	595	0	1
K	0	0	0	0	0	0	6	0	0	0	593	1
L	1	0	4	0	1	0	0	1	0	1	0	592

Script recognition is a difficult task given the variation in the words for a particular script. But the results are really encouraging for building a model that can identify the script with certainty. The results obtained after the combination exceeds the reported accuracies for this certain task and hence set the new benchmark. Table 18 provides the class wise accuracy along with the overall accuracy achieved by each procedure used in the paper. It shows that the Logistic Regression classifier acting on the MLP classifier outputs provide the best result where 98.45% accuracy is obtained with an improvement of 7.05%. Results are also obtained from the feature level combination. Natural combination or concatenation of two features at a time and all three together are done. The new feature set formed in each case undergoes the same process of classification through the MLP classifier. The comprehensive results for comparison are given in the following Table 19.

Table 18. Class-wise percentage accuracy for the classifier combination methods.

Class	Abstract Level	Rank Level	Measurement Level Combination Rules			DS Theory of Evidence			Secondary Classifier		
	Majority Voting	Borda Count	Sum Rule	Product Rule	Max Rule	DS with 2 Sources	DS with 3 Sources	3-NN	Random Forest	MLP	Logistic Regression
A	89.0	93.8	91.5	94.3	85.3	94.0	91.3	95.5	96.8	96.5	97.1
B	98.3	97.0	98.2	97.3	97.8	98.8	97.3	99.5	99.5	99.8	99.6
C	99.5	97.7	99.5	99.5	99.8	100.0	99.6	99.6	99.1	99.5	99.6
D	98.3	96.7	98.8	98.5	96.6	98.3	98.6	99.1	98.8	98.8	99.0
E	98.5	77.7	99.1	77.8	94.3	98.1	91.3	99.8	99.6	99.5	99.1
F	93.5	96.0	96.0	97.0	92.6	96.6	95.3	98.3	97.5	98.1	98.1
G	92.3	93.0	94.6	96.3	92.1	93.8	92.6	95.1	97.5	95.5	96.3
H	94.5	91.0	97.0	93.6	94.3	94.5	94.6	97.5	97.0	97.1	97.6
I	95.3	91.8	94.8	92.6	94.8	97.1	93.5	97.6	97.3	97.6	97.8
J	99.0	98.3	99.8	99.0	99.0	100.0	99.5	99.6	99.1	99.5	99.3
K	94.5	97.8	98.0	98.8	94.0	98.6	97.0	99.0	98.8	99.5	99.0
L	94.3	95.0	97.2	98.1	94.1	94.3	96.6	98.6	98.6	98.6	98.8
Overall	95.6	94.3	97.8	95.7	94.6	97.0	95.6	98.3	98.3	98.3	98.5

Table 19. Final combination results at feature level.

Feature/Methodology	Recognition Accuracy (%)
MLG	91.42
HOG	78.04
Elliptical	79.57
MLG + HOG	86.03
HOG + Elliptical	86.57
MLG + Elliptical	93.44
MLG + HOG + Elliptical	91.03
Best result after classifier combination	98.45

5. Conclusions

This is the first application of classifier combination approaches in the domain of script recognition considering the number of scripts being undertaken and the range of classifier combination procedures that are evaluated. Combination is performed at the feature level as well as decision level using abstract level, rank level and measurement level information provided by the classifiers. Encouraging results are obtained from the experiments. High accuracies in the range of 95–98% have been achieved by using combination techniques as shown in the previous Result section. There is an increase of over 7% with the best performing MLP classifier when Logistic Regression is used as the secondary classifier for 7200 samples from 12 different scripts. So, this model proves to be useful for this complex pattern recognition problem and makes a better decision based on the information provided by the base classifier.

Though, in the present work, three sources of information with different feature sets have been combined using their respective classifier results but this process can be extended to include more input sources along with different classifier. With the increase in the number of sources, an intelligent and dynamic selection procedure needs to be employed in order to facilitate combination in a more meaningful way. The combination being an overhead to the classification task, it is important to develop methods that can indicate if the combination would work or not qualitatively. In future, the work can be extended for a larger dataset so that the robustness of the procedures can be established. The script recognition system here is a general framework which can be applied to other similar pattern recognition tasks like block and line level recognition of scripts to establish its usefulness in document analysis research.

Acknowledgments: The authors are thankful to the Center for Microprocessor Application for Training Education and Research (CMATER) and Project on Storage Retrieval and Understanding of Video for Multimedia (SRUVM) of Computer Science and Engineering Department, Jadavpur University, for providing infrastructure facilities during progress of the work. The authors of this paper are also thankful to all those individuals who willingly contributed in developing the handwritten *Indic* script database used in the current research.

Author Contributions: Anirban Mukhopadhyay and Pawan Kumar Singh conceived and designed the experiments; Anirban Mukhopadhyay performed the experiments; Anirban Mukhopadhyay and Pawan Kumar Singh analyzed the data; Ram Sarkar and Mita Nasipuri contributed reagents/materials/analysis tools; Anirban Mukhopadhyay and Pawan Kumar Singh wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript and in the decision to publish the results.

References

1. Singh, P.K.; Sarkar, R.; Nasipuri, M. Offline Script Identification from Multilingual Indic-script Documents: A state-of-the-art. *Comput. Sci. Rev.* **2015**, *15–16*, 1–28. [[CrossRef](#)]
2. Ubul, K.; Tursun, G.; Aysa, A.; Impedovo, D.; Pirlo, G.; Yibulayin, T. Script Identification of Multi-Script Documents: A Survey. *IEEE Access* **2017**, *5*, 6546–6559. [[CrossRef](#)]
3. Spitz, A.L. Determination of the script and language content of document images. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 234–245. [[CrossRef](#)]
4. Tan, T.N. Rotation Invariant Texture Features and their use in Automatic Script Identification. *IEEE Tran. Pattern Anal. Mach. Intell.* **1998**, *20*, 751–756. [[CrossRef](#)]
5. Hochberg, J.; Kelly, P.; Thomas, T.; Kerns, L. Automatic script identification from document images using cluster-based templates. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 176–181. [[CrossRef](#)]
6. Hochberg, J.; Bowers, K.; Cannon, M.; Keely, P. Script and language identification for hand-written document images. *IJDAR* **1999**, *2*, 45–52. [[CrossRef](#)]
7. Wood, S.; Yao, X.; Krishnamurthi, K.; Dang, L. Language identification for printed text independent of segmentation. In Proceedings of the International Conference on Image Processing, Washington, DC, USA, 23–26 October 1995; pp. 428–431.

8. Chaudhuri, B.B.; Pal, U. An OCR system to read two Indian language scripts: Bangla and Devnagari (Hindi). In Proceedings of the 4th IEEE International Conference on Document Analysis and Recognition (ICDAR), Ulm, Germany, 18–20 August 1997; pp. 1011–1015.
9. Pal, U.; Sinha, S.; Chaudhuri, B.B. Word-wise Script identification from a document containing English, Devnagari and Telgu Text. In Proceedings of the -Second National Conference on Document Analysis and Recognition, PES, Mandya, Karnataka, India, 11–12 July 2003; pp. 213–220.
10. Chaudhury, S.; Harit, G.; Madnani, S.; Shet, R.B. Identification of scripts of Indian languages by Combining trainable classifiers. In Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing, Bangalore, India, 20–22 December 2000; pp. 20–22.
11. Das, A.; Ferrer, M.; Pal, U.; Pal, S.; Diaz, M.; Blumenstein, M. Multi-script vs. single-script scenarios in automatic off-line signature verification. *IET Biom.* **2016**, *5*, 305–313. [[CrossRef](#)]
12. Diaz, M.; Ferrer, M.A.; Sabourin, R. Approaching the Intra-Class Variability in Multi-Script Static Signature Evaluation. In Proceedings of the 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 1147–1152.
13. Padma, M.C.; Vijaya, P.A. Global Approach for Script Identification using Wavelet Packet Based Features. *Int. J. Signal Process. Image Process. Pattern Recognit.* **2010**, *3*, 29–40.
14. Hiremath, P.S.; Shivshankar, S.; Pujari, J.D.; Mouneswara, V. Script identification in a handwritten document image using texture features. In Proceedings of the 2nd IEEE International Conference on Advance Computing, Patiala, India, 19–20 February 2010; pp. 110–114.
15. Pati, P.B.; Ramakrishnan, A.G. Word level multi-script identification. *Pattern Recognit. Lett.* **2008**, *29*, 1218–1229. [[CrossRef](#)]
16. Dhanya, D.; Ramakrishnan, A.G.; Pati, P.B. Script identification in printed bilingual documents. *Sadhana* **2002**, *27*, 73–82. [[CrossRef](#)]
17. Chanda, S.; Pal, S.; Franke, K.; Pal, U. Two-stage Approach for Word-wise Script Identification. In Proceedings of the 10th IEEE International Conference on Document Analysis and Recognition (ICDAR), Barcelona, Spain, 26–29 July 2009; pp. 926–930.
18. Pal, U.; Chaudhuri, B.B. Identification of different script lines from multi-script documents. *Image Vis. Comput.* **2002**, *20*, 945–954. [[CrossRef](#)]
19. Pal, U.; Sinha, S.; Chaudhuri, B.B. Multi-Script Line identification from Indian Documents. In Proceedings of the 7th IEEE International Conference on Document Analysis and Recognition (ICDAR), Edinburgh, UK, 6 August 2003; pp. 880–884.
20. Singh, P.K.; Chatterjee, I.; Sarkar, R. Page-level Handwritten Script Identification using Modified log-Gabor filter based features. In Proceedings of the 2nd IEEE International Conference on Recent Trends in Information Systems (ReTIS), Kolkata, India, 9–11 July 2015; pp. 225–230.
21. Singh, P.K.; Sarkar, R.; Nasipuri, M.; Doermann, D. Word-level Script Identification for Handwritten Indic scripts. In Proceedings of the 13th IEEE International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, 23–26 August 2015; pp. 1106–1110.
22. Singh, P.K.; Das, S.; Sarkar, R.; Nasipuri, M. Line Parameter based Word-Level Indic Script Identification System. *Int. J. Comput. Vis. Image Process.* **2016**, *6*, 18–41. [[CrossRef](#)]
23. Nadal, C.; Legault, R.; Suen, C.Y. Complementary algorithms for the recognition of totally unconstrained handwritten numerals. In Proceedings of the 10th International Conference on Pattern Recognition, Atlantic City, NJ, USA, 16–21 June 1990; Volume A, pp. 434–449.
24. Suen, C.Y.; Nadal, C.; Mai, T.; Legault, R.; Lam, L. Recognition of totally unconstrained handwritten numerals based on the concept of multiple experts. In Proceedings of the International Workshop on Frontiers in Handwriting Recognition, Montreal, QC, Canada, 2–3 April 1990; pp. 131–143.
25. Ho, T.K. A Theory of Multiple Classifier Systems and Its Application to Visual Word Recognition. Ph.D. Thesis, State University of New York, Buffalo, NY, USA, 1992.
26. Ho, T.K.; Hull, J.J.; Srihari, S.N. Decision combination in multiple classifier systems. *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, *16*, 66–75.
27. Xu, L.; Krzyzak, A.; Suen, C. Methods of combining multiple classifiers and their applications to handwritten recognition. *IEEE Trans. Syst. Man Cybern.* **1992**, *22*, 418–435. [[CrossRef](#)]
28. Mandler, E.; Schuerman, J. *Pattern Recognition and Artificial Intelligence*; North-Holland: Amsterdam, The Netherlands, 1988.

29. Lee, D. A Theory of Classifier Combination: The Neural Network Approach. Ph.D. Thesis, State University of New York, Buffalo, NY, USA, 1995.
30. Singh, P.K.; Mondal, A.; Bhowmik, S.; Sarkar, R.; Nasipuri, M. Word-level Script Identification from Multi-script Handwritten Documents. In *Proceedings of the 3rd International Conference on Frontiers in Intelligent Computing Theory and Applications (FICTA)*; Springer: Cham, Switzerland, 2014; AISC Volume 1, pp. 551–558.
31. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
32. Daugman, J.G. Uncertainty relation for resolution in space, spatial-frequency and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am.* **1985**, *2*, 1160–1169. [[CrossRef](#)]
33. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*; Prentice-Hall: India, 1992; Volume I.
34. Singh, P.K.; Sarkar, R.; Nasipuri, M. Correlation Based Classifier Combination in the field of Pattern Recognition. *Comput. Intell.* **2017**. [[CrossRef](#)]
35. Tulyakov, S.; Jaeger, S.; Govindaraju, V.; Doermann, D. Review of classifier combination methods. In *Machine Learning in Document Analysis and Recognition*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 361–386.
36. Kittler, J. Combining Classifiers: A Theoretical Framework. *Pattern Anal. Appl.* **1998**, *1*, 18–27. [[CrossRef](#)]
37. Van Erp, M.; Vuurpijl, L.G.; Schomaker, L. An Overview and Comparison of Voting Methods for Pattern Recognition. In *Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR-8)*, Niagara-on-the-Lake, ON, Canada; 2002; pp. 195–200.
38. Kittler, J.; Hatef, M.; Duin, R.; Matas, J. On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 226–239. [[CrossRef](#)]
39. Shafer, G. *A Mathematical Theory of Evidence*; Princeton University Press: Princeton, NJ, USA, 1976.
40. Yager, R.R. On the Dempster-Shafer framework and new combination rules. *Inf. Sci.* **1987**, *41*, 93–137. [[CrossRef](#)]
41. Basu, S.; Sarkar, R.; Das, N.; Kundu, M.; Nasipuri, M.; Basu, D.K. Handwritten Bangla Digit Recognition Using Classifier Combination through DS Technique. In *Proceedings of the 1st International Conference on Pattern Recognition and Machine Intelligence (PReMI)*, Kolkata, India, 20–22 December 2005; pp. 236–241.
42. Shoyaib, M.; Abdullah-Al-Wadud, M.; Chae, O. A Skin detection approach based on the Dempster-Shafer theory of evidence. *Int. J. Approx. Reason.* **2012**, *53*, 636–659. [[CrossRef](#)]
43. Ni, J.; Luo, J.; Liu, W. 3D Palmprint Recognition Using Dempster-Shafer Fusion Theory. *J. Sens.* **2015**, *2015*, 7. [[CrossRef](#)]
44. Singh, P.K.; Chowdhury, S.P.; Sinha, S.; Eum, S.; Sarkar, R. Page-to-Word Extraction from Unconstrained Handwritten Document Images. In *Proceedings of the 1st International Conference on Intelligent Computing and Communication (ICIC2)*; Springer: Singapore, 2016; AISC Volume 458, pp. 517–524.
45. Zhang, B.; Srihari, S.N. Class-wise multi-classifier combination based on Dempster-Shafer theory. In *Proceedings of the 7th IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV 2002)*, Singapore, 2–5 December 2002; Volume 2.

