*Sādhanā* Vol. 27, Part 1, February 2002, pp. 83–97. © Printed in India

# Neural network based system for script identification in Indian documents

S BASAVARAJ PATIL and N V SUBBAREDDY*†

Kuvempu University Research Centre, Department of Computer Science and Engineering, University B D T College of Engineering, Davangere 577 004, India
†Present address: Department of Computer Science & Engineering, Manipal Institute of Technology, Manipal 576 119, India
e-mail: sbpati9@hotmail.com; dr_nvsreddy@rediffmail.com

**Abstract.** The paper describes a neural network-based script identification system which can be used in the machine reading of documents written in English, Hindi and Kannada language scripts. Script identification is a basic requirement in automation of document processing, in multi-script, multi-lingual environments. The system developed includes a feature extractor and a modular neural network. The feature extractor consists of two stages. In the first stage the document image is dilated using $3 \times 3$ masks in horizontal, vertical, right diagonal, and left diagonal directions. In the next stage, average pixel distribution is found in these resulting images. The modular network is a combination of separately trained feedforward neural network classifiers for each script. The system recognizes $64 \times 64$ pixel document images. In the next level, the system is modified to perform on single word-document images in the same three scripts. Modified system includes a pre-processor, modified feature extractor and probabilistic neural network classifier. Pre-processor segments the multi-script multi-lingual document into individual words. The feature extractor receives these word-document images of variable size and still produces the discriminative features employed by the probabilistic neural classifier. Experiments are conducted on a manually developed database of document images of size $64 \times 64$ pixels and on a database of individual words in the three scripts. The results are very encouraging and prove the effectiveness of the approach.

**Keywords.** Document processing; optical character recognition; script identification; probabilistic neural network; multi-script multi-lingual document.

## 1. Introduction

With the recent emergence and widespread application of multimedia technologies, there is increasing demand to create a paperless environment in our daily life. In transformation

*For correspondence

from the traditional paper-based society to a truly paperless electronic information society, document image processing in general and optical character recognition (OCR) in particular will play an important role. Machine reading of optically scanned text is usually called optical character recognition (Tan 1996).

Research on OCR has not been limited to any specific language/script. The recognition of characters in documents in a variety of languages (e.g. English, Chinese, Japanese, Korean, Hindi, Arabic etc.) has been addressed. Over the years a great number of OCR techniques have been developed and surveys of existing techniques may be found in several articles (Mantas 1986; Govindan & Shivaprasad 1990; Mori *et al* 1992; Muthusamy *et al* 1994; Jain & Zhong 1996; ). Almost all existing work on OCR makes an important implicit assumption that the language or script of the document to be processed is known beforehand. In an increasingly popular multilingual environment, such an assumption implies human intervention in identifying the language of documents. This is clearly undesirable (Tan 1998). The key part in the automation of document processing in this type of environments is script/language identification (Muthusamy *et al* 1994; Hochberg *et al* 1997). Script/language identification has implications both in the selection of proper character recognition service and in the resolution of errors produced by character recognition (Spitz 1997).

Existing literature on written language recognition is very limited (Tan 1998). Recently Hochberg (Hochberg *et al* 1997) described automatic script identification based on characteristic shapes or symbols of different scripts. In a training phase, cluster analysis is used to discover frequent character or word shapes in each script and a representative template is defined for each cluster. To identify the script used in a new document, the system compares a subset of the document's textual symbols to these templates and chooses the script whose templates provide best match. A "textual symbol" is any connected component in a document image that meets certain size requirements. For acceptable classification, at least 50 textual symbols are verified.

The latest work in this area is presented by Spitz (1997), in which a system based on character shape codes is described. The basis of this distinction is that upward concavities which are distributed evenly along the vertical axis of Asian characters tend to appear at certain locations in Roman characters. Further distinctions among Asian scripts are then made on the basis of character density. These distinguishing characteristics were found through hands-on analysis, a similar analysis would be required for any additional script. The classification method was based on a statistical classifier, linear discriminant analysis. Here at least two lines of text are considered to determine the script. The error rate for 4 lines of samples was 0.5%. Error rate for 2 lines of samples was 0.2% and for 1 line was 15.8%.

Very recently Tan (1998) called the above two methods 'local approaches' as they require analysis of individual components (attributes) and followed a global texture-based approach to classify 6 language 128x128 pixel document scripts. The summary of the major literature in the related work is presented in table 1.

In the context of Indian language document analysis, major literature is due to Chaudhuri and Pal (Pal & Chaudhuri 1997, 1999; Chaudhury & Sheth 1999). This group worked on automatic separation of words from multiscript documents. The method is based on hands-on analysis of individual scripts. Such analysis is required for each additional script and the method is not trainable.

Interest in neural networks is rapidly growing and several neural network models have been proposed for solving various difficult problems, especially classification problems.

**Table 1.** Summary of the major literature in related work.

| Researcher | Database size | No.of scripts/ languages | *Recommended document size | Classification technique | Features Used |
|---|---|---|---|---|---|
| Tan (1998) | 150 | 06 | 128 × 128 pixels | Weighted Euclidean | Texture |
| Spitz (1997) | 755 | 23 | 6 lines | Linear discriminate | Upward concavities & optical densities |
| Hochberg (1997) | 268 | 13 | 50 textual symbols | Template matching on Hamming distance | Characteristic shapes or symbol templates |

* Recommended size of input pattern for classification of the scripts

Traditional classifiers test the competing hypothesis sequentially, whereas neural network classifiers test the competing hypothesis in parallel, thus providing high computational speeds (Subba Reddy & Nagabhushan 1998). In this paper we propose a neural network based system for script identification. Using a neural network for solving this particular problem is a new approach. In order to take advantage of the learning and generalization abilities of the neural classifiers we propose an automatic script identification system built on a very simple neural network classifier, which has the ability to detect the script of a single word. Script identification of a single word is necessary for automation of multi-script, multi-lingual documents (Nagy 2000), has been rarely attempted and as yet is an unsolved problem of document image analysis.

The organization of the paper is as follows. As a first part, in § 2, we describe feature extraction. In § 3, we discuss the implementation of script identification system developed using a combination of neural classifiers. In § 4, we present the results of experiments conducted on this system using a single feedforward neural network classifier. In § 5, we propose a modified system to identify the script of a single word in a multi-script, multi-lingual document. Conclusions and discussions are presented in § 6.

## 2. Feature extraction

Feature extraction is an integral part of any recognition system. The aim of feature extraction is to describe the pattern by means of a minimum number of features or attributes that are effective in discriminating among pattern classes. In the presently considered problem of classification of English, Hindi and Kannada document image patterns, black pixel distribution in each script can effectively be used as a potential feature. Pixel distribution is a characteristic of a script and if it is considered along some specific directions it can discriminate different pattern classes.

A feature extraction method is used to reduce the bitmap image of a sample pattern corresponding to pixel distribution into a vector of real numbers required for classification. To produce the feature vector for an input document pattern we adopted the two-stage procedure as follows. In the first stage, each of the documents is morphologically dilated in horizontal, vertical, right-diagonal and left-diagonal directions, using $3 \times 3$ masks

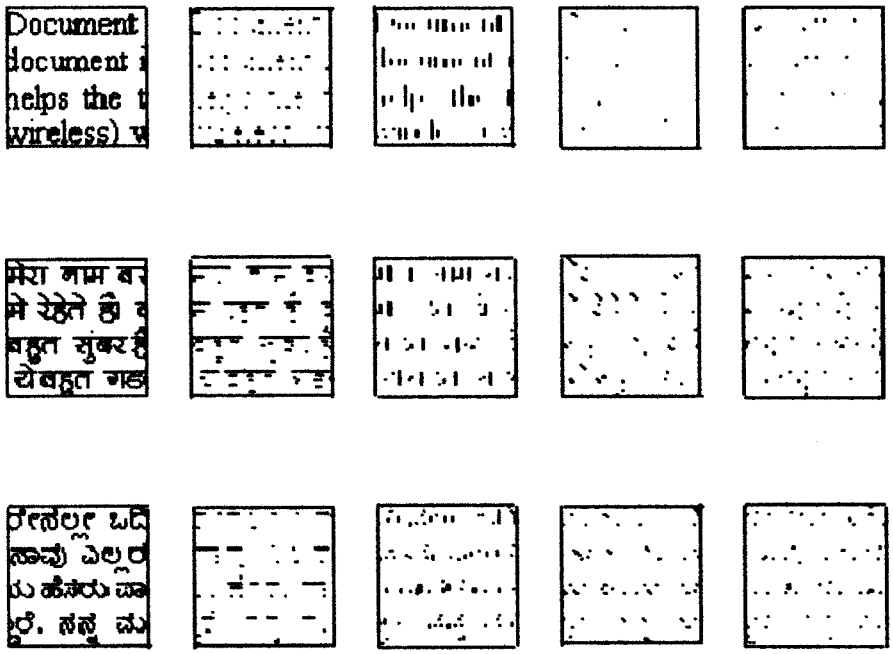

**Figure 1.** Document image samples in three scripts and the results obtained by morphological modifications using $3 \times 3$ masks, in horizontal, vertical, left diagonal, and right diagonal directions.

(Phillips 1995). The results of this method on three document samples, one each in three scripts, are shown in figure 1.

In the next stage, these four modified versions of the image and the original are used for feature extraction. The number of pixels in each of these resulting images is counted. A feature value is given by the number of marked bits (pixels) divided by the total number of pixels in the image. The total number of pixels in the image is a constant value (i.e., $64 \times 64 = 4096$). Thus a set of five features is obtained for each document image.

The results of the above method on 300 document image samples, normalized between $-1$ and $+1$ are shown in figure 2. In figure 2, 1 to 100, 101 to 200, 201 to 300, on the $x$-axis correspond to feature values obtained from 100 documents in each of English, Hindi and Kannada language scripts respectively. From the figure it is evident that the method adopted produced distinguishing feature values. The scatter plot of two sample features is shown in figure 3. This also confirms the approach.

## 3. Developed system

The language of a document is reflected in the image of that document in at least two ways: the script or the character set and writing convention of the language (Spitz 1997). Presently three languages, English, Hindi and Kannada, are considered and hence script identification is sufficient to classify the document by language.

The system architecture developed is shown in figure 4. It consists of the already discussed feature extractor module and the modular neural network. The modular

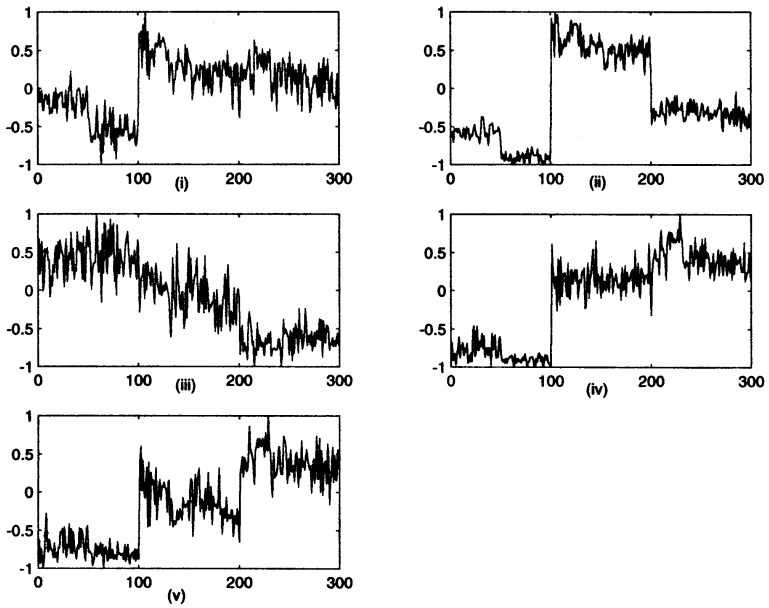


**Figure 2.** The plot of feature values for 300 document images, 100 each in English, Hindi and Kannada languages. **(i)** to **(v)** correspond to feature numbers 1 to 5.

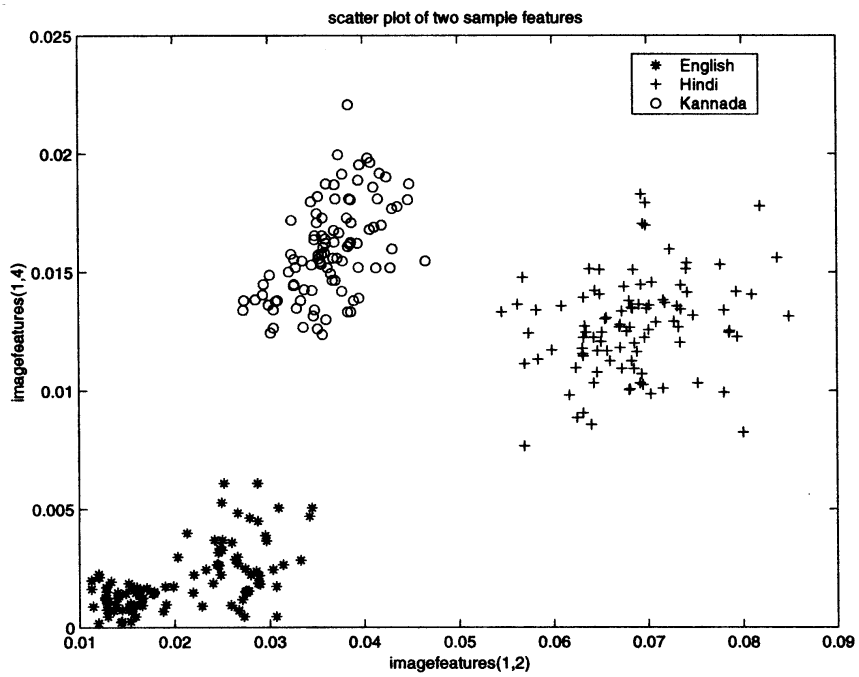


**Figure 3.** Scatter plot of two sample features.

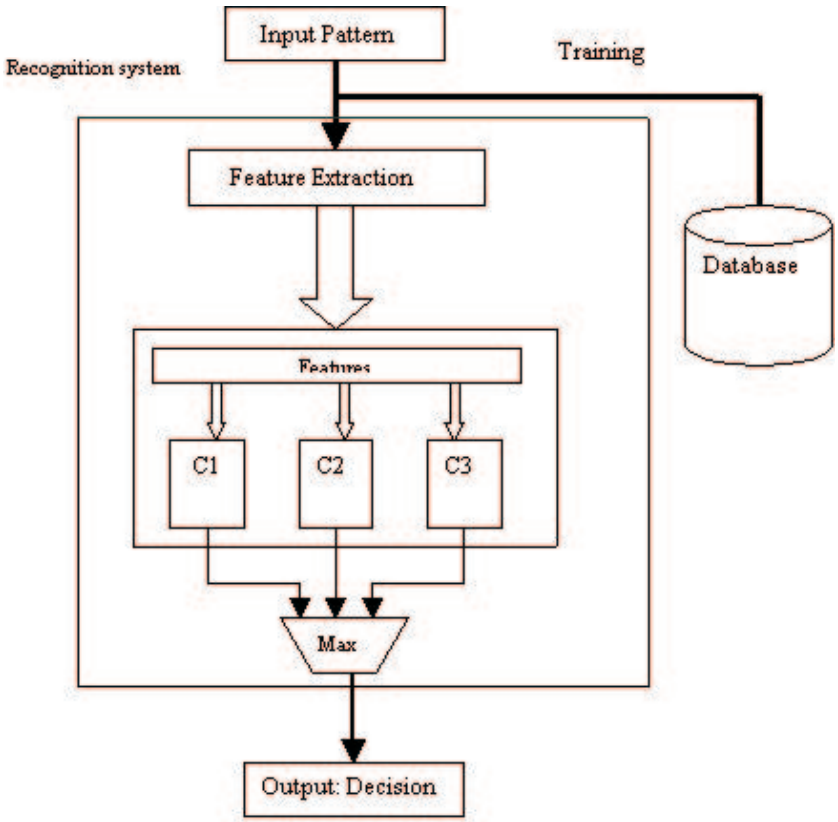


**Figure 4.** Developed system. C1, C2, and C3 are component networks.

neural network consists of three classifiers C1,C2 and C3 to classify the three scripts. Each classifier is an independently trained feedforward component neural network. We chose the feedforward neural network because of its simplicity and also since we felt it is adequate for the current problem. Each individual component network consists of 5 input nodes, 5 hidden nodes and 2 output nodes: A (Accept node) and V (Veto node). We selected the log sigmoid function for the hidden and output layer neurons.

We trained each component network to recognize only one class of scripts. That is, to train the first component network to recognize English scripts, we set the target outputs as $(A, V) = (1, 0)$ for only English patterns and for all other patterns the target outputs are set to (0, 1). Similarly, the other two component networks are trained to recognize Hindi and Kannada documents. We adopted the simple backpropagation training algorithm, with momentum and variable learning rate.

The last part of the system is the decision module. The class of the unknown document input is decided based on the output values of the three component networks. The class of the classifier, which produces maximum output at accept node, is judged as the class of the unknown input pattern. And for that classifier the output at accept node is compared with the output of veto node. If the output of the veto node is greater than that of accept node that particular sample is rejected.

**Table 2.** Matrix showing the results of document image script identification by single feedforward neural network.

| Language/script | No. of documents | Recognized as | | | Errors +rejected |
|---|---|---|---|---|---|
| | | English | Hindi | Kannada | |
| English | 100 | 99 | – | – | 01 |
| Hindi | 100 | 01 | 94 | 02 | 06 |
| Kannada | 100 | 03 | – | 95 | 05 |

## 4. Experiments and results

In the absence of any standard database, we prepared the database of 300 typed documents of size 64 × 64 pixels, 100 each in the English, Hindi, and Kannada languages.

Choice of the size of document patterns depends indirectly on the strength of feature extraction and classification techniques adopted. The block of text obviously cannot be very small, in order to ensure that there is sufficient data for reliable feature extraction (Tan 1998). Size block of text 64 × 64 pixels is chosen in our experiments to show that it is possible to classify the documents even at such reduced size of sample pattern by our developed system.

We wrote the documents in paragraphs of size around eight lines in MS-Word, and in ported these to the MS-Paint program. In the MS-Paint we manually divided documents into non-overlapping 64 × 64 pixels and saved them as two-tone document images. Sample documents are shown in figure 1. We used simple Times New Roman font of size 11 for English. We followed the same procedure to produce Hindi and Kannada document images but used the commercially available Shree-lipi fonts to write the documents. Hindi documents are written in Shree-726S00 font of size 11 and Kannada documents in Shree-854S00 font of size 11.

The above database is divided into the training and test sets. The training set includes 150 documents, 50 documents in each class and the test set includes the remaining 150 (other than those used in training). Two experiments are conducted on this database. The first experiment is on a single feedforward network, with 5 input nodes, 5 hidden nodes and 3 output nodes. Each output node corresponds to the one of three different classes: English, Hindi and Kannada. The network is trained with a simple backpropagation training algorithm, with momentum and a variable learning rate. In this experiment we fixed a threshold to accept the obtained result at 0.7. The results obtained are tabulated in table 2.

**Table 3.** Matrix showing the results of document image script identification by modular neural network.

| Language/script | No. of documents | Recognized as | | | Errors +rejected |
|---|---|---|---|---|---|
| | | English | Hindi | Kannada | |
| English | 100 | 100 | – | – | 00 |
| Hindi | 100 | 03 | 97 | – | 03 |
| Kannada | 100 | – | – | 100 | 00 |

**Table 4.** Details of single FFNN and classifiers of developed system .

| Classifier | Training time (s) | Epochs | Classification accuracy (%) |
|---|---|---|---|
| Single FFNN | 59.14 | 1863 | 96.00 |
| English | 34.578 | 1091 | 99.67 |
| Hindi | 29.958 | 924 | 96.67 |
| Kannada | 53.155 | 1781 | 100.00 |

The second experiment is conducted on the modular neural network based system. Each classifier network is trained independently, using the above mentioned training algorithm. As already discussed the target outputs are only two (1, 0) for the designated class and (0, 1) for other remaining classes. The results are tabulated in table 3.

The learning parameters for single feedforward network in the first experiment and for all the three networks in the second experiment, are kept as: learning rate at 0.01, ratio to increase learning rate at 1.05, ratio to decrease learning rate at 0.7, maximum performance increase at 1.04, momentum constant at 0.9, and mean square error goal at 0.01.

Training times, number of epochs, and classification accuracies for the above single network and modular network are as shown in table 4. The values are the averages of ten trials and are obtained on a Pentium-III 550MHz processor. The performance comparison of both the networks is made in terms of recognition rate, substitution rate, reliability and classification accuracy as shown in table 5.

## 5. Modified system

Document processing in Indian environments has a special significance, since eighteen official languages are in use. Throughout the country, every government office uses at least two languages. In several offices, three languages, namely English, Hindi and the regional language of the state, are in use. In all such environments a single document (e.g. a passport application form, a bank challan, bus or railway reservation form etc.) may contain words in two or more language scripts. The necessity of script identification in all such documents is at a word-by-word level. For automation of such documents, there should be a mechanism whereby the language of the input word is first identified and the appropriate OCR module is then selected.

To suit such environments we propose the modified system shown in figure 5. The modified system consists of pre-processor, modified feature extractor and probabilistic neural network classifier, which select the appropriate pre-processor before OCR. English,

**Table 5.** Performance comparison of the networks for script identification of document images.

| Network type | Recognition rate (%) | Substitution error (%) | Reliability (%) | Classification accuracy (%) |
|---|---|---|---|---|
| Single FFNN | 96.00 | 4.0 | 96.00 | 96.00 |
| Modular NN | 98.00 | 2.0 | 98.0 | 99.00 |

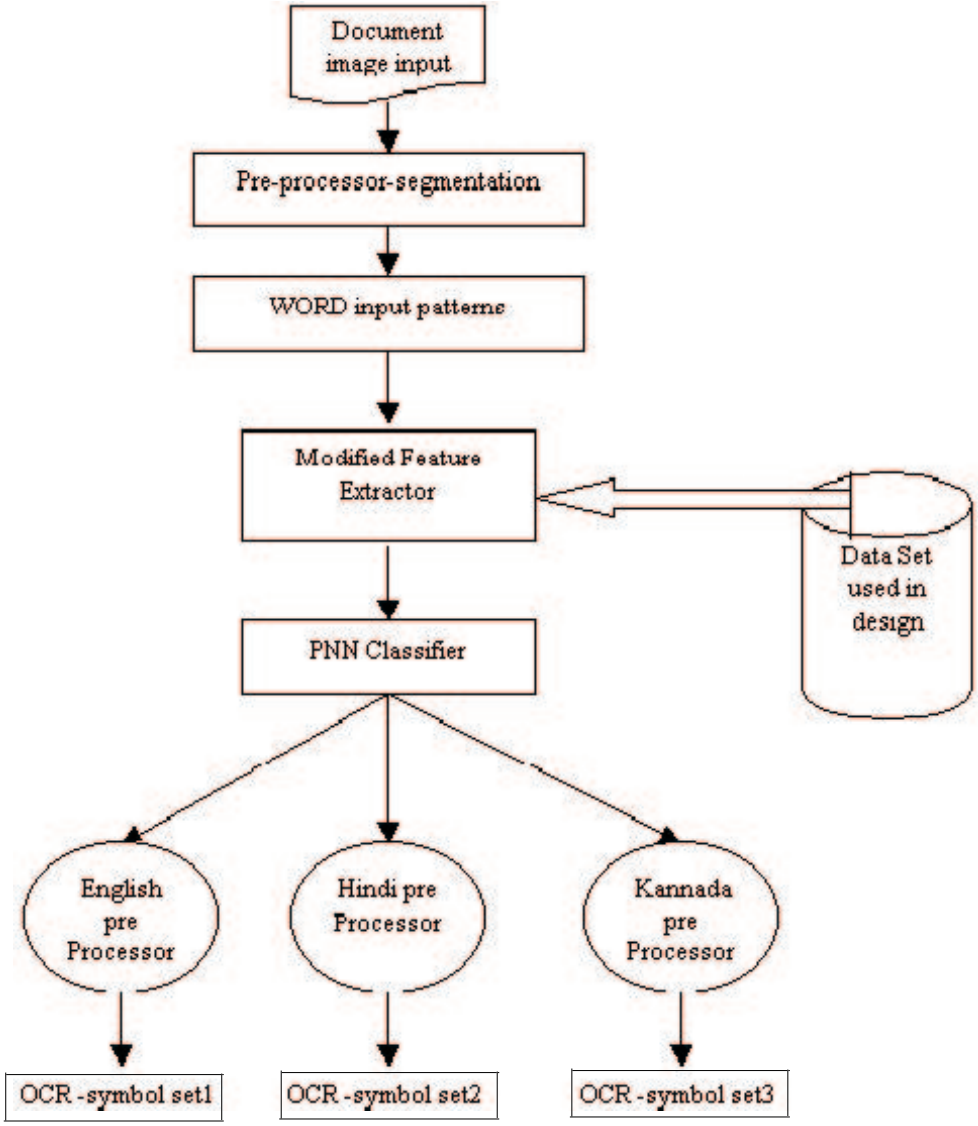


**Figure 5.** Modified system.

Hindi and Kannada pre-processors and OCR symbol sets shown in the diagram are for better understanding of the function of the proposed system and are not parts of the present system. These are script-specific and may perform further segmentation of words into characters before optical character recognition.

### 5.1 *Pre-processor*

The pre-processor used in the system receives the document written in English, Hindi and Kannada scripts, and segments into individual words using simple black to white

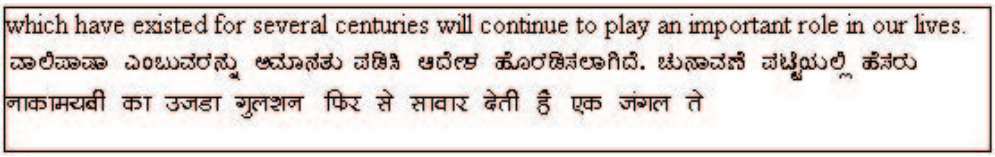


**Figure 6.** Sample multi-script document.

and white to black transitions. The sample results of the segmentation of the document shown in figure 6 are as shown in figure 7.

### 5.2 *Modified feature extractor*

The feature extractor discussed in § 2 receives an input in the form of a bitmap of size 64x64 pixels and extracts the features. But to test the script of a word, the same technique fails to calculate the features, because of the variability in the length and height of the word. This problem is solved in the modified feature extractor as follows.

The modified feature extractor receives the input word and performs the first stage operations as discussed in § 2. The results of the first stage of feature extraction on three sample words in three scripts are as shown in figure 8.

In the next stage, these four modified versions of the image and the original are used for feature extraction. Each such image is divided into ten regions, five in the horizontal direction and five in the vertical direction. The number of pixels in each region is counted. A feature value is given by the number of marked bits (pixels) in the corresponding region divided by the total number of bits in that region. By this procedure, a set of 10 features is


**Figure 7.** Results of the segmentation of the document shown in figure 6.

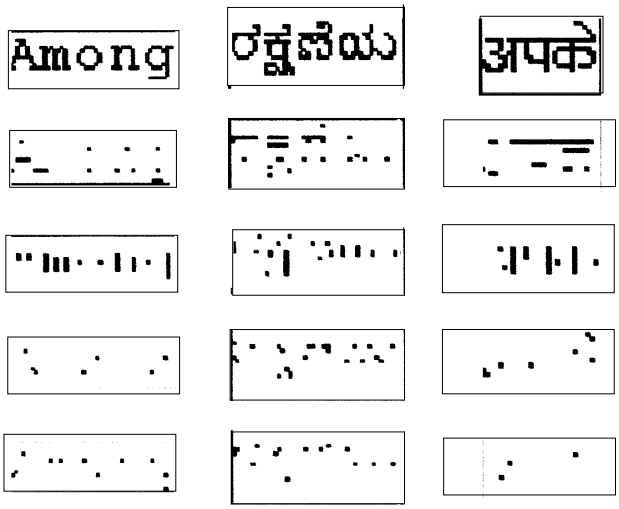


**Figure 8.** Results of the first stage of the modified feature extraction.

obtained for each output image of the first stage. Thus a set of 50 features is obtained for each word pattern. The scatter plot of randomly chosen two such features, is as shown in figure 9.

### 5.3 *Probabilistic neural network classifier*

The probabilistic neural network is a two-layered structure. The first layer is a radial basis layer and the second is a competitive layer. The first layer computes the distances

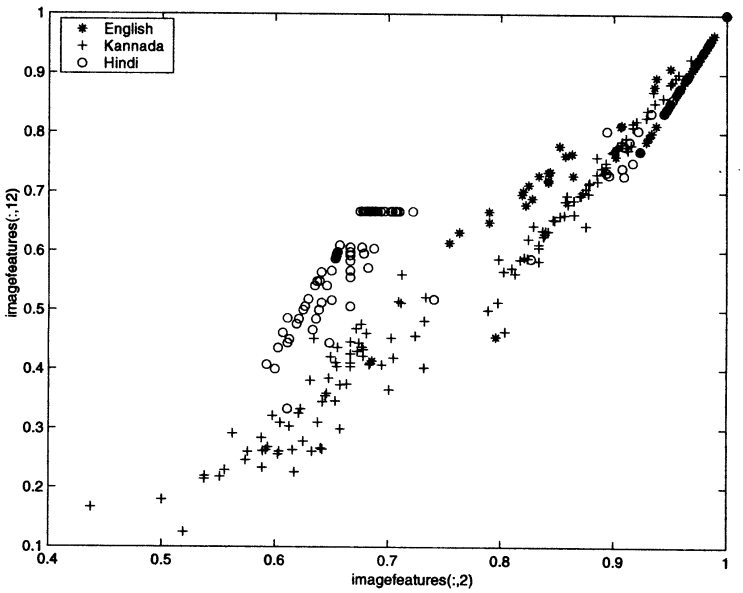


**Figure 9.** The scatter plot of two randomly chosen features.

**Table 6.** Matrix showing the results of word script identification by the modular neural network.

| Language/script | No. of words | Recognized as | | | Errors +rejected |
|---|---|---|---|---|---|
| | | English | Hindi | Kannada | |
| English | 150 | 148 | 01 | – | 02 |
| Hindi | 150 | 04 | 145 | 01 | 05 |
| Kannada | 150 | – | 02 | 148 | 02 |

from the input vector to the training input vectors and produces a vector whose elements indicate how close the input is to a training input. The second layer sums these contributions for each class of inputs to produce as its net output, a vector of probabilities (Wasserman 1993). The maximum of these probabilities is considered and the class for which it belongs is selected.

The inputs to the radial basis layer are the outputs obtained from the feature extractor module. In the reported experiments, this is a vector of size 50. This layer consists of radial basis neurons equal to the number of training patterns. (In our experiments it is 225.) The weights for this layer are set to the transpose of the matrix formed from the total number of training pairs. The net input to the radial basis neurons is the vector distance between its weight vector **w** and the input vector **p**, multiplied by bias **b**.

The output of a radial basis neuron is given by the function,

$$Y = \exp(-n^2), \tag{1}$$

where $n = \|\mathbf{w} - \mathbf{p}\|.b$ and $\|.\|$ denotes Euclidean distance. The radial basis function has a maximum of '1' when its input is 0. As the distance between **w** and **p** decreases, the output increases.

Each bias in the first layer is set to the square root $(-\log(0.5))/spread$ or $0.8326/spread$. This gives the radial basis functions that cross 0.5 at a weighted input of $+/-Spread$. This determines the width of an area in the input space to which each neuron responds. The bias $b$ allows the sensitivity of the radial basis neuron to be adjusted. For example, if a neuron has a bias 0.1, it outputs 0.5 for any input vector $p$ at a vector distance of $8.326(0.8326/b)$ from its weight vector **w**. A larger *spread* leads to a large area around the input vector, where the radial basis neurons respond with significant outputs. Therefore if *spread* is small the radial basis function is very steep so that the neuron with the weight vector closest to the input has a much larger output than other neurons. The network tends to respond with the target vector associated with the nearest design vector. In our experiments we uses trial and error method to set the *spread*.

**Table 7.** Individual classifiers of modular neural network for word script identification.

| Classifier | Training time (s) | Epochs | Classification accuracy (%) |
|---|---|---|---|
| English | 14.88 | 185 | 95.825 |
| Hindi | 48.21 | 643 | 90.447 |
| Kannada | 32.72 | 445 | 96.233 |

**Table 8.** Matrix showing the results of word script identification by simple probabilistic neural network.

| Language/script | No. of words | Recognized as | | | Errors |
|---|---|---|---|---|---|
| | | English | Hindi | Kannada | |
| English | 150 | 149 | 01 | – | 01 |
| Hindi | 150 | 02 | 148 | – | 02 |
| Kannada | 150 | – | 02 | 148 | 02 |

Competitive layer receives net input (column) vector obtained from the radial basis neurons. The number of neurons in this layer is equal to the number of classes $k$. The weights are set to the matrix **T** of target vectors. Each vector has a value one only in the row associated with that particular class of input and zeroes elsewhere. The multiplication of the matrix **T** and **Y** (column vector), sums the elements of **Y** due to each of the $k$ input classes. From this result, the second layer competitively produces 'one' corresponding to the largest element and zeroes elsewhere. Thus the network has classified the input vector into a specific one of $k$ classes because that class has the maximum probability of being correct.

### 5.4 *Experiments*

To produce a database of words in the same three language scripts, the documents in small paragraphs of size 3 to 6 lines are written in MS-word and are saved in MS-Paint as two-tone document images. These document images are segmented into individual words. By adopting the procedure repeatedly, we obtain the database of words. To create the database of words we consider documents written in a single script. From this database 150 non-similar words are selected in each script and the database of total 450 words is prepared. In the database we avoid the digits and the Hindi '|' character (denoting stop).

The above database is divided into a training set and a test set. The training set consists of 225 words, 75 words from each script class. The test set consists of the remaining 225 words (other than those used for training). Two experiments are conducted on this database. The first experiment is on a modular feedforward neural network similar to that of the second experiment in § 4. The only difference is that the classifiers are trained on word patterns and the architecture of each individual network consists of 50 nodes in the input layer, 40 nodes in the hidden layer and 2 nodes in the output layer. The other parameters of the network are the same as those in the second experiment reported in § 4. The results of this experiment are shown in table 6. The training time, epochs, and classification of individual classifiers are shown in table 7. The values are averages of ten trials.

**Table 9.** Performance comparision of the networks for script identification of words.

| Network type | Recognition rate(%) | Substitution error (%) | Reliability (%) | Classification accuracy (%) |
|---|---|---|---|---|
| Modular NN | 97.33 | 2.22 | 97.77 | 98.00 |
| PNN | 97.78 | 2.22 | 97.78 | 98.89 |

The second experiment is conducted on the probabilistic neural network based system. The spread for this network is chosen by repeated manual trials. Acceptable results are obtained at $spread = 0.17$. The results are shown in table 8.

Finally the performance comparison of both the networks is made in terms of recognition rate, substitution rate, reliability and classification accuracy as shown in table 9.

## 6. Conclusions

In this paper we propose a neural network based script identification system for English, Hindi and Kannada language documents. To the best of our knowledge, the neural network-based approach to solve script/language identification problem, proposed in this paper is the first of its kind. The feature extraction technique presented in this work is not only useful for script/language identification but also for simple or non uniform sized character recognition tasks.

As is evident from the comparison (table 4), the modular neural network performs better in script identification of document images of $64 \times 64$ pixel size. Instead of the modular neural network, other classifiers can also be used. The aim of our experiment is to show that simple neural network-based techniques are enough to solve script identification tasks. Similarly in the script identification task of single words, a probabilistic neural network performs slightly better than the modular neural network. As shown in table 8, even though the advantages of this network seem insignificant, but the advantages exist in the design. The design of a probabilistic neural network is straightforward and does not require additional training. On the other hand, modular network is a combination of three individual classifiers, where each one requires separate training.

In the experiments conducted by us, we considered three scripts/languages, English, Hindi and Kannada since such documents exist in the offices of the Karnataka State, India. However it is not a difficult task to increase the number of script classes. We can reduce the error rate in such cases by changing the feature extraction module to increase the number of features. We can expect greater accuracy if we consider only two scripts. The only limitation we felt is in procuring data sets from different scripts.

The system presented here specifically addresses the script identification issue in the automation of Indian multi-script, multi-lingual document processing. The distinct advantage of the proposed system is its ability to identify script at word level, which was not possible by earlier systems, but is a necessity for multi lingual OCR systems. The results obtained demonstrate the effectiveness of the approach followed. Other merits of the system are that it is script-independent and is able to learn with very few representative patterns.

## References

Chaudhury S, Sheth R 1999 Trainable script identification strategies for Indian languages. *5th Int. Conf. on Document Analysis and Recognition* (IEEE Comput. Soc. Press) pp 657–660

Govindan V K, Shivaprasad A P 1990 Character recognition – a review. *Pattern Recogn.* 23: 671–683

Hochberg J, Kelly P, Thomas T, Keens L 1997 Automatic script identification from document images using cluster based templates. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-19: 176–181

Jain A K, Zhong Y 1996 Page segmentation using texture analysis. *Pattern Recogn.* 29: 743–770

Mantas J 1986 An overview of character recognition methods. *Pattern Recogn.* 19: 425–430

Mori S, Suen C Y, Yamamoto K 1992 Historical review of OCR research and development. *Proc. IEEE* 80: 1029–1058

Muthusamy Y K, Barnard E, Cole R A 1994 Reviewing automatic language identification. *IEEE Signal Process. Mag.* October: 33–41

Nagy G 2000 Twenty years of document image analysis in PAMI. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-22: 38–62

Pal U, Chaudhuri B B 1997 Automatic separation of words in multi-lingual multi-script Indian documents. *4th Int. Conf. on Document Analysis and Recognition* 2: 576–579

Pal U, Chaudhuri B B 1999 Script line separation from Indian multi-script documents. *5th Int. Conf. on Document Analysis and Recognition* (IEEE Comput. Soc. Press) pp. 406–409

Philips D 1995 *Image processing using C* (New Delhi: BPB Publications) pp 459–466

Spitz A L 1997 Determination of the script and language content of document images. *IEEE Trans. Pattern Anal. Machine Intell.* 19: 235–245.

Subba Reddy N V, Nagabhushan P 1998 A connectionist expert system model for conflict resolution in unconstrained handwritten numeral recognition. *Pattern Recogn. Lett.* 19: 161–169

Tan T N 1996 Written language recognition based on texture analysis. *Proc. of IEEE Image Processing*, pp 185–188

Tan T N 1998 Rotation invariant texture features and their use in automatic script identification. *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-20: 751–756

Wasserman P D 1993 *Advanced methods in neural computing* (New York: Von Nostrand Reinhold) pp. 35–55