International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 7, July - 2013

Implementation Of Neural Network Based Script Recognition

Manisha Kumavat, Alwin Anuse
Department of Electronics and Telecommunication
Maharashtra Institute of Technology, Pune

Abstract

A variety of different scripts are used in writing different languages throughout the world. In India the documents may be printed or written in English, Hindi or any other official language. In this multi-script, multilingual environment for automatic processing of such a documents through optical character recognition (OCR), it is necessary to identify different script regions of the document. OCR systems need to be capable of recognizing characters irrespective of the script in which they are written. Recognition of different script characters in a single OCR module is difficult. Bank of OCRs corresponding to all different scripts is used. The characters in an input document can then be recognized reliably by selecting the appropriate OCR system from the OCR bank. Nevertheless this will require knowing a priori the script in which the input document is written. This paper highlights a method for automatically identifying the script used in document images. Features are extracted from pre-processed images using wavelet packet decomposition (texture based approach). Script classification performance is analyzed by using the Neural Network classifiers.

Keywords - *Document image processing, Script identification, Feature extraction, Wavelet packet decomposition, GLCM, Back propagation, GRNN, K-NN.*

1. Introduction

Documents and files that were once stored physically on paper are now being converted into electronic form in order to facilitate quicker additions, searches, and modifications, as well as to prolong the life of such records. A great proportion of business documentation and communication, however, still takes place in physical form and the fax machine remains a vital tool of communication worldwide. Because of this, there is a great demand for software which automatically extracts, analyzes, and stores information from physical documents for later retrieval. All of these tasks fall under the general heading of document analysis, which has been a fast growing area of research in recent years. One interesting and challenging field of research in pattern recognition is Optical Character Recognition (OCR). Optical character recognition is the process in which a paper document is optically scanned and then converted into computer process able electronic format by recognizing and associating symbolic identity with every individual character in the document, converting it into electronic form. To date, many algorithms have been presented in the literature to perform this task, with some of these having been shown to perform to a very high degree of accuracy in most situations, with extremely low character recognition error rates. However, such algorithms rely extensively on a priori knowledge of the script and language of the document in order to properly segment and interpret each individual character. While in the case of Latin-based languages such as English, German, and French, this problem can be overcome by simply extending the training database to include all character variations such an approach will be unsuccessful when dealing with differing script types. At best, the accuracy of such a system will be necessarily reduced. However, most OCR systems are script specific in the sense that they can read characters written in one particular script only. Script identification is an important problem in the field of document image processing, with its applications to sort document images, as pre processor to select specific OCRs, to search online archives of document images for those containing a particular language, to design a multi-script OCR system and to enable automatic text retrieval based on script type of the underlying document. Stages of document processing in a multiscript environment is illustrated in Fig. 1[2].

Script is defined as the graphic form of the writing system used to write statements expressible in language [2]. This means that a script class refers to a particular style of writing and the set of characters used in it. Languages throughout the world are type set in many different scripts. A script may be used by only one language or may be shared by many

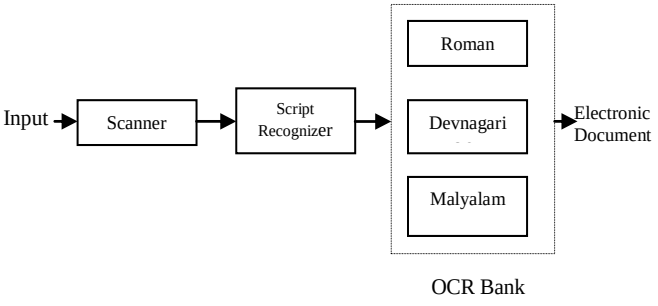

Fig.1 Stages of document processing in a multiscript environment

languages, sometimes with slight variations from one language to other. For example, Devnagari is used for writing a number of Indian languages like Sanskrit, Hindi, Konkani, Marathi, etc., English, French, German, and some other European languages use different variants of the Latin alphabet, and so on. Some languages even use different scripts at different point of time and space. For example Sanskrit, which is mainly written in Devnagari in India but is also written in Sinhala script in Sri Lanka. Every country has their own language and script. This may or may not common to other countries. To communicate with each other we need to have a common language. English is the language that is performing that role. So most of the countries (other than Roman) use bi-script documents. In India we have a total of 12 official scripts (and 22 languages) things are more complex. So identification of the script from a document may be written with any of these 13 scripts is a very challenging work. This is because every country uses its own national language and English as second/foreign language. Therefore, bi-lingual document with one language being the English and other being the national language is very common. But the things get complicated when we talk about a multi-lingual and multiscript country like India which has more than 22 official languages and 12 official scripts beside Roman (English). So here, an official document page may be written in any of the above mentioned 13 scripts. Things get complicated when an official document page may be written in more than one script. For example in a government office the possible scripts could be English, the States official language and Devnagari. So to have an OCR we need to identify the script by which the script the document is written (even the document is not itself multi-script). Postal document, pre-printed forms are good example of such multi-lingual/script documents. Therefore, in this multilingual and multiscript world, OCR systems need to be capable of recognizing characters irrespective of the script in which they are written. In general, recognition of different script characters in a single OCR module is difficult. This is because the features necessary for character recognition depend on the structural properties, style, and nature of writing, which generally differs from one script to another. For example, features used for recognition of English alphabets are, in general, not good for recognizing Chinese logograms. Fig. 2 shows block diagram for script Recognition system Fig. 2

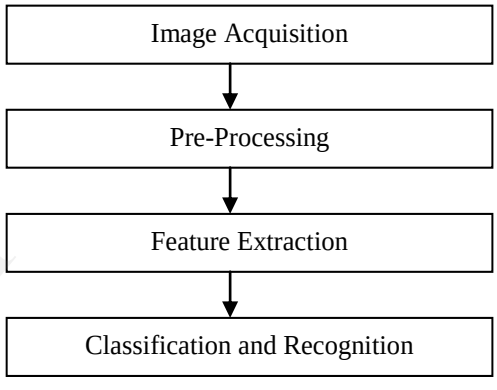

Fig. 2 Block diagram for script Recognition

2. Preprocessing

The raw data, depending on the data acquisition type, is subjected to a number of preliminary processing steps to make it usable in the descriptive stages of character analysis. Preprocessing aims to produce data that are easy for the script recognition systems to operate accurately.

Noise Removal: With non-linear filters, the noise is removed without any attempts to explicitly identify it. The median filter is one of the most popular nonlinear filters. The noise is removed by replacing the window center value by the median value of center neighborhood.

Binarization: The documents were initially digitized by a flatbed scanner in adequate (200/300/400 dpi) resolution. The digital images captured in gray tone were binarized by Otsu or Souvola algorithm's depending on the image quality. Otsu method was used for freshly printed documents of good quality. For older document pages having local intensity variations, Sauvola approach has been used. More

complex documents needed a mixture of global and local approach for proper binarization. In order to reduce storage requirements and to increase processing speed, it is often desirable to represent gray-scale or color images as binary images by picking a threshold value.

Normalization: The images must be of the same size, resolution, orientation, and scale. Line and word spacing, character sizes and heights, and the amount of white space surrounding the text, if any, can also affect texture features. In order to minimize the effects of such variations to provide a robust texture estimate, our system attempts to normalize each text region before extracting texture features. This process will remove text regions that are too small to be characterized adequately by texture features.

3. Feature extraction

3.1 Wavelet Packet Transform (WPT):

Wavelet-based methods continue to be powerful mathematical tools and offer computational advantage over other methods for texture classification. The different wavelet transform functions filter out different range of frequencies [1] (i.e. sub bands). Thus, wavelet is a powerful tool, which decomposes the image into low frequency and high frequency sub band images. The Continuous Wavelet Transform (CWT) is defined as the sum over all time of the signal multiplied by scaled, shifted versions of the wavelet function ψ:

$$C(Scale, Postion) = \int_{-\infty}^{\infty} f(t)\varphi(Scale, Position, t)dt$$

The results of the CWT are many wavelet coefficients C, which are functions of scale and position. The wavelet transform decomposes a signal into a series of shifted and scaled versions of the mother wavelet function. Due to time frequency localization properties, discrete wavelet and wavelet packet transforms have proven to be appropriate starting point for classification tasks. In the 2-D case, the wavelet transform is usually performed by applying a separable filter bank to the image. Typically, a low filter and a band pass filter are used. The convolution with the low pass filter results in the approximation image and the convolutions with the band pass filter in specific directions result in the detail images as shown in Fig. 3.

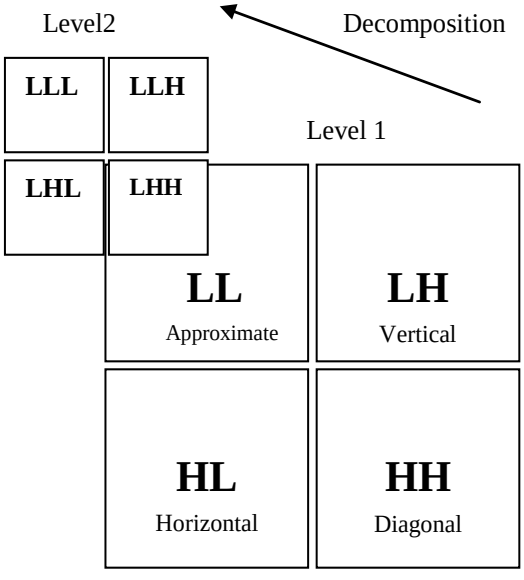

Fig. 3 Wavelet 2- level decomposition structure

The input images are decomposed through the Wavelet Packet basis function to get the four sub band images namely Approximation (A) and three detail coefficients - Horizontal (H), Vertical (V) and the Diagonal (D) as shown in Fig. 4. The Haar wavelet transformation is chosen because the resulting wavelet bands are strongly correlated with the orientation elements in the GLCM computation. The second reason is that the total pixel entries for Haar wavelet transform are always minimum. Haar basis function up to level two is used in this method. This result in a total of 20 sub bands [1].

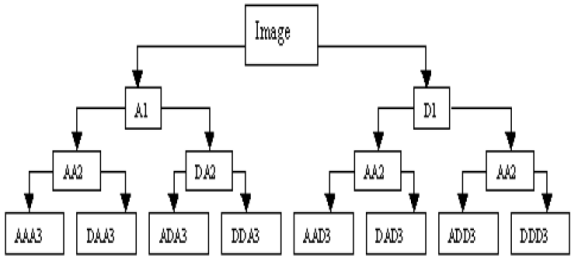

Fig. 4 Wavelet Packet Decomposition Tree

Group 1:
Approximation sub bands: (1, 0), (2, 0) = (A, AA)
Group 2:
Horizontal sub bands: (1, 1), (2, 1), (2, 4), (2, 5) = (H, AH, HA, HH)
Group 3:
Vertical sub bands: (1, 2), (2, 2), (2, 8), (2, 10) = (V, AV, VA, VV)
Group 4:
Diagonal sub bands: (1, 3), (2, 3), (2, 12), (2, 15) = (D, AD, DA, DD)

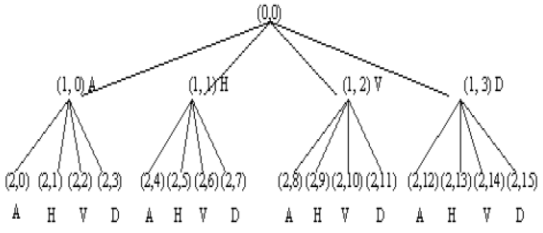

Fig.5 Wavelet Packet Tree up to Level-2

Thus, only fourteen sub bands - two approximate sub band, four horizontal sub band, four vertical sub bands and four diagonal sub bands are selected out of the twenty sub bands from Fig.5. The wavelet packet coefficients are quantized using the quantization function

$$q(x) = round\,(k\ log\left(\frac{x}{a_s\partial}+1\right)+1/2)$$

Where $k = I - 1/\log\left(\frac{1}{\partial}+1\right)$Then, gray level co-occurrence matrices are constructed for the quantized wavelet sub bands [7].

3.1.1 Gray Level Co-occurrence Matrices (G LCMs)

GLCM is a two dimensional measure of texture, which show how often each gray occurs at a pixel located at a fixed geometric position relative to each other pixel, as a function of its gray level. For the approximate sub bands i.e., group1 ((1, 0), (2, 0)), GLCMs are constructed with the value θ = {0^0, 45^0, 90^0, 135^0}. The value of θ is taken as 0^0 for horizontal sub bands (group2), 90^0 for vertical sub bands (group3) and, 45^0 and 135^0 for diagonal sub bands (group4). Thus, totally, twenty four GLCM (eight GLCM for group1, four GLCM for group2, four GLCM for group3 and eight GLCM for group4) are constructed[1].

4. Wavelet Packet Co-occurrence Features Extracted from a Co-occurrence Matrix C(i, j)

Inertia

$$F1 = \sum_{i,j=0}^{n}(i-j)^2 C(i,j)$$

Total Energy

$$F2 = \sum_{i,j=0}^{n} C^2(i,j)$$

Entropy

$$F3 = -\sum_{i,j=0}^{n} C(i,j)\log C(i,j)$$

Contrast

$$F4 = -\sum_{i,j=0}^{n} C(i,j)|i-j|^k$$

Local Homogeneity

$$F5 = \sum_{i,j=0}^{n} 1/(1+(i-j)^2)\ C(i,j)$$

Cluster Shade

$$F6 = \sum_{i,j=0}^{n}\left(i-M_x+j-M_y\right)^3 C(i,j)$$

Cluster Prominence

$$F7 = \sum_{i,j=0}^{n}\left(i-M_x+j-M_y\right)^4 C(i,j)$$

$$M_x = \sum_{i,j=0}^{n} iC(i,j) \qquad M_y = \sum_{i,j=0}^{n} jC(i,j)$$

The eight Haralick texture features are extracted from the twenty four GLCM resulting in a total of 168 features [1].

5. Classification

5.1 Back Propagation Neural Network:

The steps for training of back propagation:

Step 1: Initialize weights, learning parameter and the performance goal (MSE).

Step 2: The extracted features are given as input to the Back Propagation network.

Step 3: The input training pattern is fed forward to obtain the actual output.

Step 4: From the actual output, the associated error is calculated and back propagated.

Step 5: The weights are adjusted accordingly and the same process is repeated.

Step 6: Repeat steps 2-5 till the performance goal is reached or the total number of epochs are completed.

The steps for testing of Script:

Step 1: The features are to be recognized are given as input to the neural network.

Step 2: The neural network is already trained to recognize the given Script.

Step 3: Back Propagation will classify the new input into the group of the same character trained earlier.

Step 4: This means the Script has been recognized.

5.2 Generalized Regression Neural Network

The steps for training of Generalized Regression Neural Network:

Step 1: Initialize sigma value i.e. spread factor.

Step 2: The extracted features are given as input to the Generalized Regression Neural network.

Step 3: The input training pattern is fed forward to GRNN.

Step 4: Network is created.

The steps for testing of Script:
Step 1: The features are to be recognized are given as input to the GRNN.
Step 2: The neural network is already trained to recognize the given Script.
Step 3: GRNN will classify the new input into the group of the same character trained earlier.
Step 4: This means the Script has been recognized.

5.3 K-Nearest Neighbor:

Step 1: Determine parameter K = number of nearest neighbors.
Step 2: Calculate the distance between the query-instance and all the training samples.
Step 3: Sort the distance and determine nearest neighbors based on the K-th minimum distance.
Step 4: Gather the category Y of the nearest neighbors.
Step 4: Use simple majority of the category of nearest neighbors as the prediction value of the query instance.

6. Testing And Results

6.1 Database

Data which is to be processed are document images of a postal document written in Hindi or English. The scanning of these documents is done at 300dpi or 200dpi. These documents contain lot of variability in terms of font size, styles and scanning resolutions varying. Fig.6, Fig 7 & Fig. 8 shows postal document written in English, Hindi & Malyalam[1].

i. English

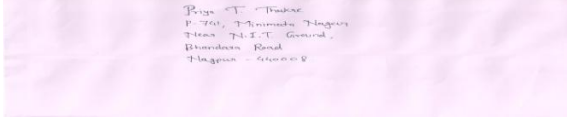

Fig.6 Postal document in English

ii. Hindi

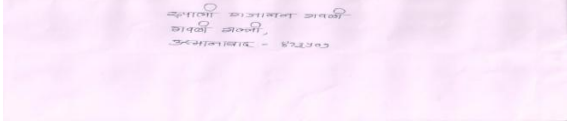

Fig. 7 Postal document in Devnagari

iii. Malyalam

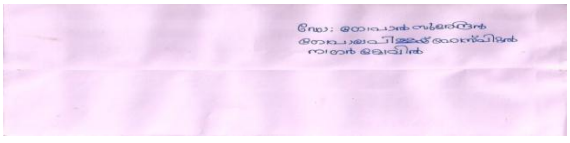

Fig. 8 Postal document in Malyalam

6.2 Recognition rates for Three Scripts Devnagari, Malyalam & English

Table 1 shows recognition rates for three scripts unseen images scanned at 300dpi by using Backpropagation

Sr. No.	No. of Neurons Used	Accuracy
1	10	61.7%
2	15	70.2%
3	25	61.7%

Table 1: Recognition Rates by Backpropagation

Table 2 shows recognition rates for *three* scripts unseen images scanned at 300dpi by using Generalized Regression Neural Network

Sr. No.	Spread	Accuracy
1	0.05	71%
2	0.07	61.7%
3	0.1	66%
4	0.2	58%
5	0.7	34%

Table 2: Recognition Rates by GRNN

Table 3 shows recognition rates for *three* scripts unseen images scanned at 300dpi by using K-Nearest Neighbor.

Sr. No.	K	Accuracy
1	1	80%
2	3	68%
3	5	64%

Table 3: Recognition Rates by K-NN

6.3 Confusion matrix

	English	Hindi	Malyalam
English	85.7%	7.15%	7.15%
Hindi	0%	85.7%	14.3%
Malyalam	5.9%	11.7%	82.4%

6.4 Performance Plot

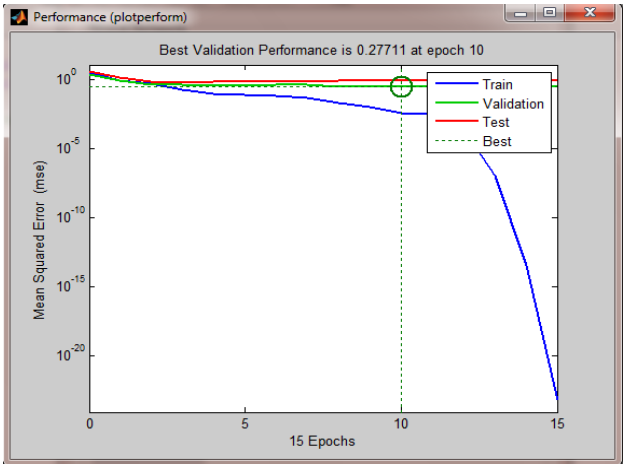

Fig.9 Performance Plot for back propagation of Images scanned at 300dpi

7. Conclusion

In this paper, a texture-based global approach is presented which can identify two scripts using a new set of texture features. The texture features are extracted from the GLCMs constructed from a set of wavelet packet sub band coefficients. Thus, the proposed global approach of script identification in a document image facilitates many important applications such as separating a huge collection of documents printed in different scripts for further processing like selecting the script specific OCR system in a multilingual environment. Another application of the proposed method is that the method can be extended to identify and separate more number of script classes as the script independent features are used. Hence, the proposed global approach has the potential to become a generalized approach for script identification.

8. References

[1] M.C. Padma and P.A.Vijaya, "Global Approach for Identification using Wavelet Packet Based Features" International journal of SP IPPR, vol-3, No-3, Karnataka, India-2010.

[2] Debashis Ghosh, Tulika Dube and Adamane P. Shivaprasad, "Script Recognition –A review" IEEE transaction on PAMI, Vol. 32, No. 12, December-2010.

[3] M. C. Padma and P.A.Vijaya, "Script Identification of Text words from a Tri Lingual Document Using Voting Technique" International journal of IP, Vol-4, Issue-1, Karnataka, India.

[4] K. Roy, S. Kundu Das and Sk Md Obaidullah, "Script Identification from Handwritten Document" Third National Conference on PR, IP & G, Kolkata-2011.

[5] Oirvind Due Trier, Anil K Jain and Torfinn Taxt, "Feature Extraction Methods for Character Recognition" Pattern Recognition, Vol.29, No.4, pp.641-662, USA- 1996.

[6] Hiremath P. S., Shivashankar S., Jagdeesh D. Pujari and V. Mouneswara, "Script identification in a handwritten document image using texture features", IEEE 2nd International Advance Computing Conference, Karnataka -2010.

[7] Andrew Busch, Wageeh W. Boles and SridhaSridharan," Logarithmic Quantisation of Wavelet Coefficient for Improved Texture Classification Performance", IEEE Conference Publications, Vol-3, Page no. 569-72, May 2004.

[8] Donald F. Specht, "A General Regression Neural Network", IEEE Transactions On Neural Networks. Vol. *2*. No. 6. November- 1991.

[9] Andrew Busch, Wageeh W. Boles and Sridha Sridharan, "Texture for Script Identification" IEEE transaction on PAMI, Vol. 27, No. 11, November 2005.

[10] Saharkiz, "K Nearest Neighbor Algorithm Implementation and Overview", An Article, 4 Feb 2009.

[11] Gang Sun, Steven Hoff, Brian Zelle, Minda Nelson, "Development and Comparison of Backpropagation and Generalized Regression Neural Network Models to Predict Diurnal and Seasonal Gas and PM10 Concentrations and Emissions from Swine Buildings" An ASABE Meeting Presentation, Paper No. 085100 Rhode Island, 2 July -2008.

[12] Nafiz Arica and Fatos T. Yarman-Vural, "An Overview of Character Recognition Focused on Off-Line Handwriting", IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 31, No. 2, May- 2001.