Header Based Classification of Journals Using Document Image Segmentation and Extreme Learning Machine

Kalpana S *kalpana.msccs@gmail.com*
Research Scholar
PSGR Krishnammal College for Women
Coimbatore, India.

Vijaya MS *msvijaya@grgsact.com*
Associate Professor
PSGR Krishnammal College for Women
Coimbatore, India.

Abstract

Document image segmentation plays an important role in classification of journals, magazines, newspaper, etc., It is a process of splitting the document into distinct regions. Document layout analysis is a key process of identifying and categorizing the regions of interest in the scanned image of a text document. A reading system requires the segmentation of text zones from non-textual ones and the arrangement in their correct reading order. Detection and labelling of text zones play different logical roles inside the document such as titles, captions, footnotes, etc. This research work proposes a new approach to segment the document and classify the journals based on the header block. Documents are collected from different journals and used as input image. The image is segmented into blocks like heading, header, author name and footer using Particle Swarm optimization algorithm and features are extracted from header block using Gray Level Co-occurrences Matrix. Extreme Learning Machine has been used for classification based on the header blocks and obtained 82.3% accuracy**.**

Keywords: Classification, Document Segmentation, Feature Extraction, Extreme Learning Machine.

1. INTRODUCTION

In computer vision, document layout analysis is the process of identifying and categorizing the regions of interest in the scanned image of a text document. Document image segmentation is a process of subdividing the document into distinct regions or blocks. It is important process in the document analysis. Document segmentation is a fundamental step in document processing, which aims at identifying the relevant components in the document that deserve further and specialized processing. Document analysis consists of geometric and logical analysis. In geometric based segmentation, the document is segmented upon its geometric structure such as text and non-text regions. Whereas in logical segmentation the document is segmented upon its logical labels assigned to each region of the document such as title, logo, footnote, caption, etc., [1]. The geometric layout analysis is also called as physical layout analysis. The physical layout of a document refers to the physical location and boundaries of various regions in the document image.

The process of document layout analysis aims to decompose a document image into a hierarchy of homogenous regions such as figures, backgrounds, text blocks, text lines, words, characters, etc., Logical structure is the result of dividing and subdividing the content of a document into increasingly smaller parts on the basis of the human-perceptible meaning of the content [2]. A logical object is an element of the specific logical structure of a document. For logical objects no classification other than basic logical objects, composite logical objects and document logical

root. The logical objects, which are the subject of extraction in the proposed method, are roughly categorized into the following headlines, headers, footers, captions, notes, and programs, titles, paragraphs, lists, and formulas.

Document layout analysis algorithms can be categorized into three approaches namely top-down approaches, bottom-up approaches and hybrid approaches. Top-down algorithms start from the whole document image and iteratively split it into smaller ranges. Bottom-up algorithms start from document image pixels, and cluster the pixels into connected components such as characters which are then clustered into words, lines or zones. Hybrid algorithms can be regarded as a mix of the above two approaches. The Docstrum algorithm was presented in [3], the Voronoi-diagram-based algorithm was proposed in [4] the run-length smearing algorithm was implemented in [5] and the text string separation algorithm is implemented by [6] are typical bottom-up algorithms. The X − Y cut-based algorithm of [7] and the shape-directed-covers-based algorithm [8] are top-down algorithms. In [9] the author proposed a hybrid algorithm using a split-and-merge strategy. The advantage of using top-down approach is, its high speed processing and the drawback is, it cannot process table, improper layout documents and forms.

This research work proposes the document segmentation based on logical layout. The segmentation of document image is done using Particle Swarm Optimization (PSO). The document image is segmented as header, heading, footer, author name. From the segmented blocks, features are extracted using Gray Level Co-occurrence Matrix (GLCM), which is the statistical method of examining the textures that considers the spatial relationship of the pixels. Features such as Energy, Entropy Autocorrelation, Contrast, Correlation, Cluster Prominence, Cluster Shade, Dissimilarity, Homogeneity, Maximum probability, Variance, Sum average, Sum variance, Sum entropy, Difference variance, Difference entropy, Information measure of correlation1 and correlation2 are computed. Energy and entropy are renowned properties of an image. Energy identifies the uniformity of the image and entropy identifies the randomness of the image. Finally the classification is performed based on header block of the document image using Extreme Learning Machine.

2. PROPOSED MODEL FOR DOCUMENT SEGMENTATION AND CLASSIFICATION

The proposed work aims to segment the document image based on logical layout. For this the documents are collected from five different journals and they are used as the input. First the noise is removed from the given input document image using median filter. The noiseless image is used for segmenting the document using the Particle Swarm optimization (PSO) algorithm and the features are extracted. The features are extracted using Gray Level Co-occurrence Matrix (GLCM). At last, the classification of journals based on the header block is carried out by using Extreme Learning Machine and Support Vector Machine. The overview of the proposed work is shown in Fig.1.

2.1. Pre-processing

Pre-processing is a sequence of tasks performed on the image. It enhances the quality of the image for segmentation. The various tasks performed on the image in pre-processing stage are scanning of documents, binarization and noise removal.

2.1.1. Scanning of Documents

The documents are collected from various journals and only the first page of each document is scanned. They are stored in the database and used as input image.

2.1.2. Binarization

It is a process which converts the gray scale image into a binary image using the global threshold method. A binary image has only two values 0 or 1 for each pixel. 0 represents white pixel and 1 represents black.

2.1.3. Noise Removal

Filters are used to remove the noise in the image or document. The noise is removed using median filter. The segmentation is focused with the noiseless image for best result.

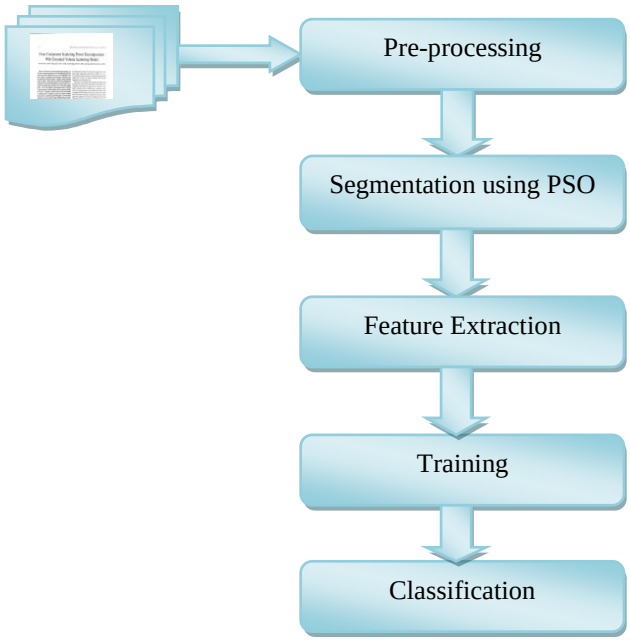

FIGURE 1: Block Diagram of Proposed methodology.

2.2. Segmentation Using Pso

The PSO algorithm is used for segmenting the document. The document is subdivided into blocks like heading, header, author name and footnote. The space between the lines is used to separate the lines. Normally the distances between two lines are larger than the distances between words, thus lines can be segmented by comparing this distance against a suitable threshold. To determine an optimal threshold, Particle Swarm Optimization technique is used. Particle Swarm Optimization (PSO) algorithm is used to solve many of difficult problems in the field of pattern recognition. Hence, PSO is used to compute an optimal value.

Let X and V denote the particle's position and its corresponding velocity in search space respectively. At iteration K, each particle i has its position defined by $X_i^k = (x_{i1}, x_{i2}, \dots, x_n)$ and a velocity is defined by $V_i^k = (v_{i1}, v_{i2}, \dots, v_{in})$ in search space n. Velocity and position of each particle in next iterations can be calculated using following equation (1) and (2).

$$V_{ij}^{k+1} = wv_{ij}^k + C_1 r_1 \left(pbest_{ij}^k - x_{ij}^k\right) + C_2 r_2 \left(gbest_{ij}^k - x_{iji}^k\right) \quad (1)$$

$$x_{ij}^k = x_g^k + v_g^k \quad (2)$$

Where k is the current iteration number, w is inertia weight, v_{ij} is then updated velocity on the i^{th} dimension of the j^{th} particle, C_1 and C_2 is acceleration constants, C_1 and C_2 is positive constant parameters, usually $C_1 = C_2 = 2$. r_1 and r_2 , are the real numbers drawn from two uniform random sequences of U (0, 1). The algorithm starts by generating randomly initial population of the PSO. Every particle is initialized with locations and velocities using the equations (1) and (2).These locations consist of the initial solutions for the optimal threshold. The procedure of the proposed PSO algorithm is described as follows.

Step 1: Initialize N particles with random positions $x_1, x_2, \ldots, x_n$ according to equation (2) and velocities V_i where i = 1, 2...N.

Step 2: Evaluate each particle according to equation (4)

$$f(t) = w_0(t) \times w_1(t) \times \left(\mu_0(t) - \mu_1(t)\right)^2$$

Where, t is a gray level between 0 and 255 which can be obtained through the particle's position

Step 3: Update individual and global best positions. If $f(pbest_i) < f(x_i)$, then $pbest_i = x_i$ search for the maximum value f_{max} among $f(pbest_i)$, if $\max f(gbest) < f_{max}$ then $gbest = x_{max}$, x_{max} is the particle associated with f_{max}.

Step 4: Update velocity: update the i^{th} particle velocity using the equation (2) restricted by maximum and minimum threshold v_{max} and v_{min}.

Step 5: Update Position: update the i^{th} particle position using equation (2) and (3).

Step 6: Repeat step 2 to 5 until a given maximum number of iterations is achieved or the optimal solution so far has not been improved for a given number of iteration.

By performing the above steps the header block is segmented for a document image for the classification of journals is shown in Figure.2.

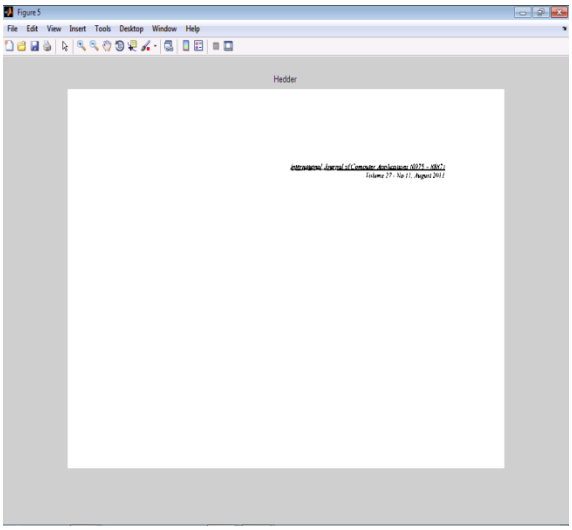

FIGURE 2: Segmented Header.

2.2.1. Labeling Connected Components

With a selection of optimal threshold value, the connected areas will form blocks of the same region. Labeling is the process of identifying the connected components in an image and assigning each component a unique label an integer number which must be same as connected black runs. Figure.3. shows the labeled connected components of document image.

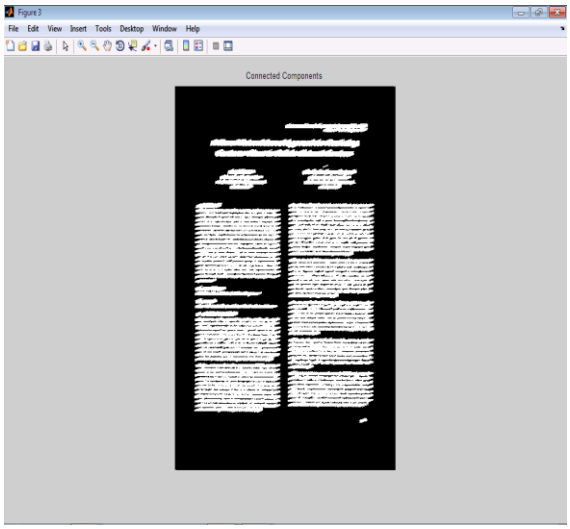

FIGURE 3: Labeling Connected Components.

2.3. Feature Extraction

Features such as Energy, Entropy Autocorrelation, Contrast, Correlation, Cluster Prominence, Cluster Shade, Dissimilarity, Homogeneity, Maximum probability, Variance, Sum average, Sum variance, Sum entropy, Difference variance, Difference entropy, Information measure of correlation1 and correlation2 are computed for classification of document region[10]. Few of the common statistics applied to co-occurrence probabilities are discussed below.

1) Energy

This is also called uniformity or angular second moment. It measures the textural uniformity that is pixel pair repetitions. It detects disorders in textures. Energy reaches a maximum value equal to one.

2) Entropy

This statistic measures the disorder or complexity of an image. The entropy is larger when the image is not texturally uniform and many GLCM elements have very small values. Complex textures tend to have high entropy.

3) Contrast

It measures the spatial frequency of an image and difference moment of GLCM. It is the difference between the highest and the lowest values of a contiguous set of pixels. It measures the amount of local variation present in the image.

4) Variance

It is a measure of heterogeneity and is strongly correlated to first order statistical variable such as standard deviation. Variance increases when the gray level values differ from their mean.

5) Homogeneity

If weights decrease away from the diagonal, the result will be larger for windows with little contrast. Homogeneity weights values by the inverse of the contrast weight, with weights decreasing exponentially away from the diagonal.

6) Correlation

The correlation feature is a measure of gray tone linear dependencies in the image. GLCM correlation is quite a different calculation from the other texture measures. It also has a more intuitive meaning to the actual calculated values: 0 is uncorrelated, 1 is perfectly correlated.

7) Autocorrelation

An autocorrelation function can be evaluated that measures the coarseness. This function evaluates the linear spatial relationships between primitives. If the primitives are large, the function decreases slowly with increasing distance whereas it decreases rapidly if texture consists of small primitives. However, if the primitives are periodic, then the autocorrelation increases and decreases periodically with distance.

The rest of the textural features are secondary and derived from those listed above.

8) Sum Variance

$$\text{sum variance (sv)} = \sum_{i=2}^{2N_g} (i - sa)^2 g_{x+y}(i)$$

9) Difference variance

$$\text{difference variance} = \text{variance of } g_{x-y}$$

10) Sum Average

$$\text{sum average (sa)} = \sum_{i=2}^{2N_g} i g_{x+y}(i)$$

11) Sum Entropy

$$\text{sum entropy (se)} = -\sum_{i=2}^{2N_g} i g_{x+y}(i) \log\{g_{x+y}(i)\}$$

12) Difference Entropy

$$\text{difference entropy} = -\sum_{i=0}^{N_g-1} g_{x-y}(i) \log\{g_{x-y}(i)\}$$

13) Information Measures of Correlation

i) Information Measures of Correlation 1 (IMC1)

$$IMC1 = \frac{HXY - HXY1}{\max\{HX, HY\}}$$

ii) Information Measures of Correlation 2 (IMC2)

$$IMC2 = \sqrt{(1 - \exp[-2.0(HXY2 - HXY)]}$$

14) Cluster shade

$$\text{Shade} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{i + j - \mu_x - \mu_y\}^3 * P(i,j)$$

15) Cluster Prominence

$$\mathrm{Prom} = \sum_{i=0}^{G-1} \sum_{j=0}^{G-1} \{i + j - \mu_x - \mu_y\}^4 * P(i,j)$$

16) Dissimilarity

$$\mathrm{Diss} = \sum P_{i,j} * |i - j|$$

Dissimilarity is a measure that defines the variation of grey level pairs in an image. It is the closest to contrast with a difference in the weight.

3. EXTREME LEARNING MACHINE

Extreme Learning Machine (ELM) is a new learning algorithm for Single-hidden Layer Feed forward neural Networks (SLFNs) supervised batch learning which provides good generalization performance for both classification and regression problems at highly fast learning speed. The output function of the generalized SLFN is given by,

$$F(x) = \sum_{i=1}^{L} \beta_i \, h_i(x)$$

Where $h_i(x)$ is the output of the i^{th} hidden- node. The ELM algorithm which consists of three steps that are: Given a training set $N = \{(x_i, t_i), x_i \in R^n, t_i \in R^m, i = 1, \ldots, N\}$, kernel function f(x), and hidden neuron $\tilde{N}$. The ELM algorithm which consists of following steps:

Step 1: Select suitable activation function and number of hidden neurons $\tilde{N}$ for the given problem.

Step 2: Assign arbitrary input weight w_i and bias $b_i, i = 1, \ldots, H$

Step 3: Calculate the output matrix H at the hidden layer

$$H = f.(w.+x + b)$$

Step 4: Calculate the output weight β

$$\hat{\beta} = H^{-1}T$$

In kernel based ELM, If the hidden layer feature mapping $h(x)$is unknown to users, users can be described a kernel function for ELM. ELM Kernel function is given by, $KELM(xi, xj) = 1/H\, f(xi).f(xj)$. That is, the data has feed through the ELM hidden layer to obtain the feature space vectors, and their co-variance is then calculated and scaled by the number of hidden units. The main difference is that where ELM explicitly generates the feature space vectors, but in SVM or another kernel method only similarities between feature space vectors are used. The entire above mentioned can be used to apply in regression, binary and multi-label classification applications directly. Kernel ELMs can be applied to complex space as well.

4. EXPERIMENTS AND RESULTS

The documents used for creating the dataset are collected from various journals like IEEE, IJCA, IJDKP, International Journal of Advances in Image Processing and European International Journal of Science and Technology. The dataset consists of 76 document images and in that 59 is used for training and remaining for testing. In the first phase each instance is segmented into four blocks as heading, header, footer and author name. After the segmentation, for each header

block 19 features are extracted using GLCM. The dataset is then trained using ELM based on the header block for the classification of journals and it is compared with SVM classifier for predictive accuracy.

The prediction accuracy and learning time of the ELM is observed. The function elm_train is used to train the model by identifying elm type, number of hidden neurons and activation function as parameters. The elm_predict function is used to test the model as [output] = elm_predict (TestingData_File). To calculate the accuracy the whole testing data is used. Based on the accuracy and the learning time the performance evaluation of the proposed work is obtained. The classification result of ELM gives the list of document headings in the specific folder based on the header block of the input image. The accuracy of ELM is evaluated using the following formula and achieved 82.3% accuracy.

$$\text{Accuracy} = \frac{\text{Number of correctly recognized image}}{\text{Total number of images in test database}} * 100$$

To compare with ELM the second experiment is carried out using the same dataset and the classification algorithm SVM is trained using the same dataset to create the classifier. The accuracy of the SVM classifier is tested using the same test dataset and the classification results are obtained as 64.7% accuracy.It is observed from the results that the performance of the model built based on Extreme Learning Machine for classification of segmented document image is more accurate and fast compared to Support Vector Machine. Comparative results of Support Vector Machine and Extreme Learning Machine are summarized in Table I. The comparative results in terms of accuracy and learning time of both classifiers are shown in Figure.4 and Figure.5 respectively.

Classifiers	Learning Time (seconds)	Accuracy (%)
SVM	15.29	64.7
ELM	13.35	82.3

TABLE 1: Comparative Results of the Classifiers.

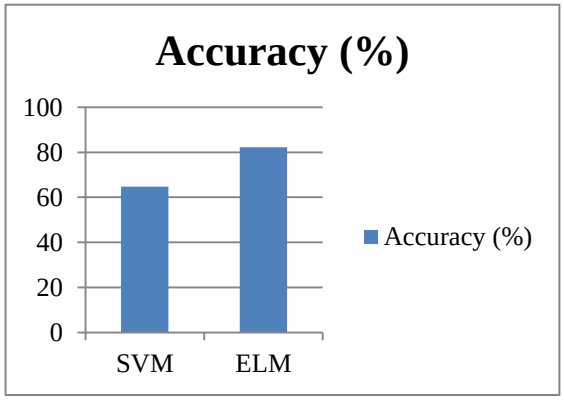

FIGURE 4: Comparison - Accuracy of Classifiers.

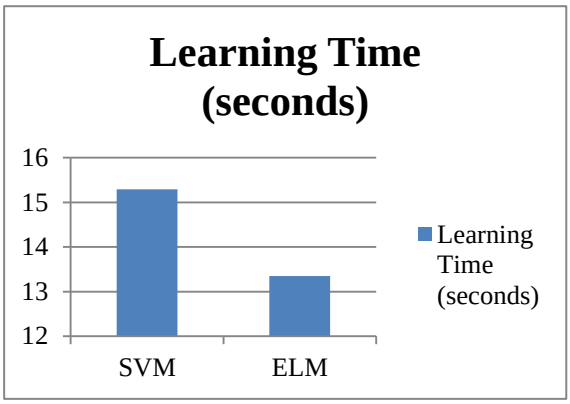

FIGURE 5: Comparison - Learning Time of Classifiers.

5. CONCLUSION

This paper demonstrates the modeling of document segmentation and classification task that describes the implementation of machine learning approach for segmenting the document into various regions. The corpus is created by collecting the documents from five different journals and stored in the database. These documents are pre-processed to remove the noise using median filter. The pre-processed documents are segmented into various blocks such as heading, header, author name and footer using Particle Swarm Optimization algorithm. From each header block the features are extracted and the training dataset is created. Finally classification based on header blocks is done using supervised classification algorithms namely ELM and SVM. The performance of both classifiers is evaluated in terms of accuracy and learning time. It has been observed that ELM technique shows better performance than SVM technique for document image classification. Future work of segmentation can be extended by detecting images, postal codes, handwritten and printed documents with more features.

6. REFERENCES

[1] Okun O. Doermann D and M. Pietikainen. ‘‘Page segmentation and zone classification’’. The state of the art. In UMD, 1999.

[2] Yuan. Y. Tang and M. Cheriet, Jiming Liu, J.N Said, “Document Analysis and recognition by computers”.

[3] L. O. Gorman, “The document spectrum for page layout analysis,” IEEE Transactions on Pattern Analysis and Machine Intelligence 15, pp. 1162–1173, 1993.

[4] K. Kise, A. Sato, and M. Iwata, “Segmentation of page images using the area Voronoi diagram,” Computer Vision and Image Understanding 70, pp. 370–382, 1998.

[5] Wahl. K. Wong, and R. Casey, “Block segmentation and text extraction in mixed text/image documents,” Graphical Models and Image Processing 20, pp. 375–390, 1982.

[6] L. A. Fletcher and R. Kasturi, “A robust algorithm for text string separation from mixed text/graphics images,” IEEE Transactions on Pattern Analysis and Machine Intelligence 10, pp. 910–918, 1988.

[7] Nagy, S. Seth, and M. Viswanathan, “A prototype document image analysis system for technical journals,” Computer 25, pp. 10–22, 1992.

[8] S. Baird, S. E. Jones, and S. J. Fortune, “Image segmentation by shape-directed covers,” in Proceedings of International Conference on Pattern Recognition, pp. 820–825, (Atlantic City, NJ), June 1990.

[9] T. Pavlidis and J. Zhou, “Page segmentation and classification,” Graphical Models and Image Processing 54, pp. 484–496, 1992.

[10] Haralick R.M., Shanmugam K., Dinstein I., “Textural Features for Image Classification”, IEEE Trans. on System Man and Cybernetics, 1973, 3(6), p.610-621.

[11] Santanu Chaudhury, Megha Jindal, and Sumantra Dutta Roy, “Model-Guided Segmentation and Layout Labeling of Document Images using a Hierarchical Conditional Random Field”, New Delhi, India.

[12] Jianying Hu, Ramanujan Kashi, Gordon Wilfong, “Document Classification using Layout Analysis”, USA.

[13] Gerd Maderlechner, Angela Schreyer and Peter Suda, “Information Extraction from Document Images using Attention Based Layout Segmentation”, Germany.

[14] Y. Ishitani. Document layout analysis based on emergent computation. Proc. 4th ICDAR, 1:45–50, 1997.

[15] K. T. Spoehr. Visual information processing. W. H. Freeman and Company, 1982.

[16] Robert M. Haralick,”Document image Understanding: Geometric and Logical layout”, University of Washington, Seattle.

[17] ISO: 8613: Information Processing-Text and Office Systems-Office, Document Architecture (ODA) and Interchange Format, International Organization for Standardization, 1989.

[18] Y. Ishitani. Logical structure analysis of document images based on emergent computation. Proc. 5th ICDAR, 1999.

[19] Esposito, F., Malerba, D., Francesca, Lisi, F.A., Ras, W.: Machine learning for intelligent processing of printed documents. Journal of Intelligent Information Systems 14 (2000) 175–198.

[20] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan, “Syntactic segmentation and labeling of digitized pages from technical journals,” IEEE Transactions on Pattern Analysis and Machine Intelligence 15, pp. 737–747, 1993.