

DOCUMENT IMAGE ANALYSIS USING INTEGRATED IMAGE AND NEURAL PROCESSING

Daniel X. Le*, George R. Thoma*, and Harry Wechsler**

*National Library Of Medicine, 8600 Rockville Pike, MS 55, Bethesda, MD 20894

**Department Of Computer Science, George Mason University, Fairfax, VA 22030

Abstract

In this paper we present robust algorithms for detecting the page orientation (portrait/landscape) and the degree of skew for binary document images, and a method for classification of binary document images into textual or non-textual data blocks using neural network models. The performance of four neural network models are compared in terms of training times, memory requirements, and classification accuracy, and it was found that the radial basis functions performed best. The experiments show the feasibility of building an integrated document analysis system for page orientation and skew angle detection, and textual block classification.

Index Terms - Page orientation, projection profiles, document skew angle, Hough transform, back propagation, radial basis functions, probabilistic neural networks, textual classification, self-organizing feature maps

1. Introduction and background

The conversion of paper-based document information to electronic format is important for any document related applications. We describe in this paper several components of the document analysis system that includes the page orientation detection, the skew angle detection, and the classification of textual and non-textual data blocks of binary document images. The page orientation and skew angle detection algorithms are robust to the presence of non-textual data. For document classification, four neural network (NN) models were considered and they include the back propagation (BP), the radial basis functions (RBF), the probabilistic neural network (PNN), and the Kohonen's self-organizing feature map (SOFM).

The *page orientation* is defined as the printing direction of text lines. The *skew angle* is defined as the orientation angle of text lines. A *block* (object) is a connected component (blob) and it is defined here as a collection of black run-lengths that are 8-connected. Textual data occurs as a letter, word, or sentence, while examples of non-

textual data include graphics, forms, line art, large fonts and dithered images.

For page orientation, the Akiyama et al. [1] technique is based on global analysis of projection profiles that is sensitive in the presence of non-textual data. The Hinds et al. [2] algorithm is based on the counts of short run-lengths in the vertical and horizontal histograms of an image. This method works well only with a document in which text data predominates. For skew angle, the Fletcher et al. [3] and Rastogi et al. [4] algorithms are both slow due to the excessive data required to be processed. The Baird [5] method works well on portrait mode layouts, but there is no discussion of its suitability to landscape images. The Hinds et al. [2] algorithm is based on Hough transform and the image resolution has to be reduced to increase the speed. However, the data reduction factor is still small. The commercial ScanFix [6] method works well with portrait images, but poorly on landscape images. For document classification, Wahl et al. [7] used a linear adaptive classification technique. The Pavlidis et al. [8] technique is based on strength and variation of correlation of adjacent scanlines. While these two techniques were reported to perform well, no information was provided on the size of the experiments and performance statistics were lacking.

2. Document analysis system

The document analysis system presented here consists of three processes as shown below.

2.1 Page orientation process

Page Orientation Process uses local analysis and the main goal is to reduce detection errors by minimizing the effects of the non-textual data on the page orientation decision. It consists of three steps as follows:

Step 1.1 is the Textual and Non-Textual Squares Classification. This step divides the entire binary image into squares where the size of a square is defined in Appendix

and categorizes these squares as textual or non-textual data squares. A non-textual square is one in which:

(1) The ratio between the total black pixels and the total pixels in a square is less than BLANKRATIO, or

(2) The ratio between the total black pixels and the total pixels in a top-half square, or in a bottom-half square, or in a left-half square or in a right-half square is less than BLANKRATIO, or

(3) The ratio between the total black pixels and the total pixels in a square is greater than GRAPHRATIO, or

(4) The ratio between the total black pixels and the total pixels in a top-half square, or in a bottom-half square, or in a left-half square or in a right-half square is greater than GRAPHRATIO, or

(5) The length of any black run in a row or in a column of a square is greater than BIGFONTSIZE, or

(6) The sum of the black runs in a row or in a column of a square is greater than MAXOBSQR, or

(7) At least the total of γ rows or columns of a square that satisfy the following condition: In each row or each column of a square, the total black pixels is less than or equal to twice the number of black runs.

Step 1.2 is the *Textual Squares Page Orientation Estimation*. It estimates the page orientation of each textual square by using (a) a projection profiles method or (b) a square difference method. Non-textual squares are not used in this step in order to reduce the impact of non-textual data on the page orientation results.

A projection profile (histogram) is obtained by mapping a binary image into a one-dimensional array. The projection profiles method determines the page orientation by analyzing the shapes of the horizontal and vertical projection histograms. Since conventional typesetting has gaps between adjacent text lines (text pattern), the shape of a projection histogram consisting of this text pattern determines the orientation.

The square difference method is another way to determine the page orientation by comparing the difference in the squared sums of the horizontal and vertical projection histograms. Empirically, the projection histogram that has more fluctuation determines the orientation and since the squared sums of the histograms represent the amount of fluctuation, the page orientation is based on the histogram having the larger squared sums.

The projection profiles method would work faster than the square difference method because it does not require a lot of calculations. However, it can be used successfully only for squares whose text is not skewed and whose text is not in fixed-pitch font. Therefore, the page orientation is first estimated using the projection profiles method. When no page orientation conclusion can be made, the square difference method must then be used. At the end of this step, each textual square is assigned its mode

weight as the total black pixels of a portrait or landscape mode textual square.

Step 1.3 is the *Squares Grouping*. This step determines the page orientation of a binary image using the notion of a pyramidal image data structure. The binary image was divided into squares in step 1.1 and these squares constitute the first layer of the pyramid. The square grouping technique then groups every 9-neighboring squares of the first layer together to build a second layer square. The page orientation and the mode weight of each second layer square are determined by comparing its total portrait mode black pixels and its total landscape mode black pixels. The larger between them decides the page orientation and mode weight of the second layer square. This squares grouping process is continued until the last 9-neighboring squares belonging to the last layer of the pyramid are computed. The last 9-neighboring squares are then used to determine the page orientation of the binary image.

2.2 Skew angle process

Skew Angle Process takes only a portion of a binary image as its input and produces a skew angle as its output. The page orientation process provides not only the page orientation but also where the textual data areas are located in the binary image. Using this information, the skew angle can be determined by selecting and processing only a small portion of the binary image consisting of textual data. The skew angle detection algorithm is based on the processing of pixels of the last black run-lengths of objects and it consist of four steps as follows:

Step 2.1 is the *Textual Area Selection and Rotation*. This step selects one of the 9 squares of the last layer of the pyramid resulting from the page orientation algorithm. The page orientation of this chosen square must be the same as that of the binary image and its mode weight is the largest among those squares having the same page orientation. Since this skew angle process works well only on a portrait mode binary image, it is required to rotate any landscape mode chosen square into a portrait mode square before continuing to the next step.

Step 2.2 is the *Component Labelling*. This step segments the chosen square into objects. Each black run-length is assigned an integer number called *label* and the labels of connected black run-lengths must be the same. Objects are then discriminated using their labels.

Step 2.3 is the *Data Reduction*. This step creates a simplified square from the chosen square by preserving only bottom pixels of candidate objects. A candidate object is an object that satisfies all of the following empirical conditions:

(1) $\text{MINWIDTH} < \text{object width} < \text{MAXWIDTH}$,

(2) MINHEIGHT < object height < MAXHEIGHT,

(3) MINAREA < object area < MAXAREA.

Bottom pixels of objects are used as feature points to determine the document skew angle. Using only candidate objects bottom pixels improve the Hough transform speed performance by reducing the amount of data to be processed, reduce the participation of the non-textual data and increase the detection accuracy rate.

Step 2.4 is the Hough Transform. In this step, the skew angle of a binary image is detected by applying the Hough transform on the simplified square.

2.3 Document classification process

Document Classification Process is based on NN and it consists of four steps as follows:

Step 3.1 is the Block Run Lengths Detection. This step segments a binary document image into blocks using the constrained run length algorithm [7].

Step 3.2 is the Block Labeling. This step is similar to the *Component Labelling* step in the previous section.

Step 3.3 is the Block Features Calculation. In this step, features of each block in a binary document image are calculated. Let BC be the number of black pixels of each block, DC be the number of black pixels of the original data within each block, and TC be the number of white-black (or black-white) transitions of the original data within each block. Also, let H and W be the height and the width of each block. Seven block features are used for classification and they are normalized to have unit length. The first four features are similar to those defined by Wahl et al. [7]: H , $E = W/H$, $S = BC/(W*H)$, and $R = DC/TC$. The last three block features are combined from the first four block features to enhance input representation and to eliminate the need for more hidden layers [9]: $HR = H*R$, $ER = E*R$, and $SR = S*R$.

Step 3.4 is the NN Learning and Classification. The training and testing data sets used in this experiment are created from fifty binary document images covering a wide variety of layouts were selected from 12 different medical journals. These fifty images consist of 129 non-textual data blocks and 2175 textual data blocks that are divided randomly into (1) 85 non-textual data blocks and 1450 textual data blocks for the *training data set* and (2) 44 non-textual data blocks and 725 textual data blocks for the *testing data set*. Furthermore, the cross-validation (CV) training technique [10] is used by randomly dividing the training data set into five data groups of which four data groups create a *CV-train set* (68 non-textual data blocks and 1160 textual data blocks) and one remaining data group is considered as a *CV-test set* (17 non-textual data blocks and 290 textual data blocks). The modified weights corresponding to the winning pair of a

CV-train set and a CV-test set, the one yielding the highest classification accuracy, are chosen to be the final weights for that particular NN. The testing data set is used to estimate the network's classification accuracy.

3. Experimental results

Experiments using images from different medical journals were run on a DELL 486D/50 Personal Computer. All documents used are 8.5 x 11 inches and scanned at 200 dpi.

3.1 Page orientation and skew angle

For the page orientation, two sets of test samples are used to conduct the experiment: (1) 6,087 pages from 63 different medical journals, and (2) 5,190 pages from 52 different medical journals. Page orientation was detected at an accuracy rate of 99.99% for both sets.

For the skew angle, two sets of test images, which are known to have many skewed images, are selected for the experiment: (1) 12 selected binary images representing both non-textual and textual data pages, and (2) 238 pages from 3 different medical journals. Skew angle was detected with an accuracy of about 0.50 degrees.

3.2 Document classification

The BP [9] is implemented with seven block feature inputs, one binary output for non-textual or textual data, and one hidden layer of which the number of nodes is 14. The size of hidden layer is calculated based on the number of inputs, outputs, and training samples as pointed out by Hush et al. [10]. The training time was about 5 hours and the best classification accuracy is about 99.4 %.

The RBF [11] is implemented with seven block feature inputs, one binary output for non-textual or textual data, and one hidden layer of which the number of nodes is the number of basis function nodes F. The means and variances of the basis functions are determined using K-means clustering algorithm. The classification accuracy depends upon H and F where H is the global proportional factor. Therefore, the RBF is also trained and tested with different values of H and F. The training time was about 47 seconds and the highest classification accuracy is about 99.6 % for $F = 10$ and $H = 150$.

The PNN [12] is implemented with seven block feature inputs, one binary output for non-textual or textual data, and one hidden layer consisting of P pattern units where P is the number of training samples. The error rate depends upon the smoothing parameter s because it controls the shape of the network exponential activation function. Therefore, in order to observe changes in its

behavior, the PNN is trained with different values of smoothing parameter s . The training time was about 14 seconds and the best classification accuracy is 98.2 % for any value of s that is greater than or equal to 1.6.

The SOFM [9] is implemented with seven block feature inputs, and an output layer - a square output matrix. Since the performance depends upon the size of the square output matrix layer, the network is trained with different values of the square output matrix size. After being trained, the resulting network is calibrated by supervised labeling of the square output matrix neuron using known vector inputs from the training data set. Each neuron in the square output matrix has two features: a label (**G** for non-textual and **T** for textual) and a response value. When a block data input vector from the training data set is presented, labels and response values of the neuron which produces the strongest response and its 8 neighborhood neurons in the square output matrix are calculated as follows: If the current neuron response is stronger than its previous response, then (a) update its response by the current response, and (b) update its label to **T** if the input vector is textual and to **G** if the input vector is non-textual. Otherwise, nothing changes. The training time was about 14 hours and the best classification accuracy is about 99.2 % when the output matrix size is 36 (1296 output neurons).

4. Conclusions

We have presented robust algorithms for detecting the page orientation and the degree of skew for binary documents images and a method for classification of binary document images into textual or non-textual data blocks using neural network models. The page orientation and skew angle detection algorithms perform very well on a variety of medical journal pages independent of text dominance. For document classification, all four neural network models including back propagation, radial basis functions, probabilistic, and Kohonen's self-organizing feature map showed very good classification results. The back propagation neural network takes more time to train, but it requires less memory. The probabilistic neural network requires that the entire training data must be stored and used for each classification of an unknown pattern. The Kohonen's self-organizing feature map requires a large memory size for the output neuron array. On the other hand, the radial basis functions neural network has the highest classification accuracy, and it requires intermediate amounts of memory and training time. We conclude that the radial basis functions neural network is the best architecture for this particular application. We are presently integrating the three processes

toward developing a smooth and efficient document analysis system.

5. References

1. T. Akiyama and N. Hagita, Automated Entry System for Printed Documents. *PR* 23(11): 1141-1154, 1990.
2. S. Hinds, J. Fisher and D. D'Amato, A Document Skew Detection Method Using Run-Length Encoding and the Hough Transform. *10th Int. Conf. on PR*, vol. 1, pp. 464-468, 1990.
3. L. Fletcher and R. Kasturi, A robust algorithm for text string separation from mixed text/graphics images. *IEEE Trans. on PAMI*, vol. 10, pp. 910-918, 1988.
4. A. Rastogi and S. Srihari, Recognizing textual blocks in document images using the Hough transform. *TR 86-01*, Department of Computer Science, SUNY Buffalo, NY, 1986.
5. H. Baird, The skew angle of printed documents. *Proc. of Soc. of Photo. Sci. and Eng.*, vol. 40, pp. 21-24, 1987.
6. Sequoia Data Corp, *ScanFix Image Opt.*, Ver 2.10 MS/DOS.
7. F. Wahl, K. Wong and R. Casey, Block Segmentation and Text Extraction in Mixed Text/Image Documents. *CGIP* 20: 375-390, 1982.
8. T. Pavlidis and J. Zhou, Page Segmentation and Classification. *CVGIP* 54(6): 484-496, 1992.
9. J. Zurada, *Introduction to Artificial Neural Systems*. West Publishing Company, St. Paul, MN, 1992.
10. D. Hush and B. Horne, Progress in Supervised Neural Networks - What's New Since Lippmann? *IEEE SP*: 8-39, 1993.
11. K. Ng and R. Lippmann, A Comparative Study of the Practical Characteristics of Neural Network and Conventional Pattern Classifiers. *TR 894*, MIT Tech., Lincoln Lab., 1991.
12. D. Specht, Probabilistic Neural Networks. *NN*:109-18,1990.

6. Appendix

In this section, empirical parameters required for the system will be defined and whenever possible formulas will be derived. Let **dpi** (dots per inch) represent the scanning resolution. The parameter values and formulas are given as follows:

Parameters	Value	For 200 dpi
Size Of A Square	$2 \times (14 \times \text{dpi}) / 72$ pixels	80 pixels
GRAPHRATIO	0.444	0.444
BLANKRATIO	0.045	0.045
BIGFONTSIZE	$(15 \times \text{dpi}) / 72$ pixels	42 pixels
MAXOBSQR	$2 \times \text{Size Of A Square} / 5$	32
γ	$\text{Size Of A Square} / 2$	40
MINWIDTH	1 pixel	1 pixel
MAXWIDTH	$(15 \times \text{dpi}) / 72$ pixels	42 pixels
MINHEIGHT	1 pixel	1 pixel
MAXHEIGHT	$(15 \times \text{dpi}) / 72$ pixels	42 pixels
MINAREA	4 pixels ²	4 pixels ²
MAXAREA	$[(15 \times \text{dpi}) / 72]^2$ pixels ²	1764 pixels ² .