Parameter Optimization for Content-based Image Enhancement

Mohamed N. Ahmed and Ahmed Eid
Software Research, Lexmark International, Inc.
Lexington, KY, USA

Abstract

In this paper, we present a content-based filtering technique to enhance scanned documents. An image classification step is performed to classify each pixel into text, background, and image regions. With the segmentation step, we can strongly sharpen text and similar edge detail while smoothing background and image content. To optimally select the segmentation parameters, we formulate a cost function to minimize the number of miss-classified pixels between classified test and reference images. This cost function is minimized using genetic algorithms.

Introduction

Document images typically contain a combination of text, background, and halftoned images. Text, including line art and similar graphical content, is characterized by sharp, high-contrast edges and thin strokes. The background of the document is usually white or nearly white, and it normally has a smooth texture. Halftoned images consist of a pattern of small dots. In most cases, the dots are arranged in an ordered pattern and will vary slightly in size according to the darkness of the image they represent.

Digital copying, in which a digital image is obtained from a scanning device and then printed, involves a variety of inherent factors that compromise image quality. Specifically, digitally copied documents usually suffer from blurring, flare, noise, and moire. To address these distortions, content-based filtering is applied to the scanned document. A segmentation step is performed to classify each pixel into text, background, and halftoned image regions. With the segmentation step, we can strongly sharpen text and similar edge detail, smooth and perhaps lighten document backgrounds, and descreen halftoned images using an appropriate low-pass filter.

Several approaches for document segmentation have been proposed [3]-[8]. These techniques can be broadly classified as bottom-up or top-down. Bottom-up methods start from the pixel level and merge regions together into larger and larger components. Top-down techniques apply a priori knowledge about the page to hypothesize and split the page into blocks which are subsequently identified and further subdivided. Top-down approaches work well with pre-specified layouts such as technical papers. However, the performance of these techniques degrades significantly when different components are touching or overlapping. Among bottom-up approaches, texture-based schemes have attracted much attention [6]-[8].

These methods treat different components of a document image as different textures. The scanned document images are convolved with a set of masks to generate feature vectors. Each feature vector is then classified into different classes using a pre-trained classifier. One problem associated with these approaches is the mask size for extracting local features. If the mask size is too small, it is difficult to detect large scale textures such as large fonts. On the contrary, if a large mask is chosen, the computational complexity will increase dramatically [8].

Due to performance and memory requirements, we propose a simple document segmentation technique that involves pairwise pixel comparison in multiple windows centered around each pixel. The result is a large number of intensity differences that are compared with appropriate thresholds to classify each pixel. The quality of the segmentation step greatly depends on the selection of these thresholds.

In this paper, we present the classification technique as well as a method to optimize the selection of the segmentation parameters. Using training images, we formulate a cost function to minimize the number of miss-classified pixels. This cost function is minimized using genetic algorithms.

Pixel Classification

In scanned documents, text, including line art and similar graphical content, is characterized by sharp, high-contrast edges and thin strokes. Background and images normally have smooth texture. Therefore, an effective way to extract text and line borders is to apply an edge detector. Edge Detection is carried out by convolving the scanned image I by a set of kernels of size 3×3 and 5×5. The kernels to extract horizontal and vertical edges are defined as follows

$$H_{3e}(i,j) = \begin{pmatrix} 0.25 \\ 0.50 \\ 0.25 \end{pmatrix} \times \begin{pmatrix} 1 & -1 & 1 \end{pmatrix} \tag{1}$$

$$V_{3e}(i,j) = \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} \times \begin{pmatrix} 0.25 & 0.5 & 0.25 \end{pmatrix} \tag{2}$$

$$H_{5e}(i,j) = \begin{pmatrix} 0.25 \\ 0 \\ 0.50 \\ 0 \\ 0.25 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & -1 & 0 & 1 \end{pmatrix} \tag{3}$$

$$V_{5e}(i,j) = \begin{pmatrix} 1 \\ 0 \\ -1 \\ 0 \\ 1 \end{pmatrix} \times \begin{pmatrix} 0.25 & 0 & 0.5 & 0 & 0.25 \end{pmatrix} \quad (4)$$

The image is first passed through a 1D mapping function $f(\cdot)$ to perform tone correction and enhance light fine details. The adjusted image is then convolved with the previous kernels to produce horizontal and vertical edge maps

$$E_1(i,j) = V_{3e}(i,j) * f(I(i,j)) \quad (5)$$

$$E_2(i,j) = H_{3e}(i,j) * f(I(i,j)) \quad (6)$$

$$E_3(i,j) = V_{5e}(i,j) * f(I(i,j)) \quad (7)$$

$$E_4(i,j) = H_{5e}(i,j) * f(I(i,j)) \quad (8)$$

where $*$ denotes convolution. An edge is declared if the absolute value output of either of the filters exceeds an appropriate threshold

$$E(i,j) = \begin{cases} 2, & |E_1(i,j)| \geq p_1 \text{ or } |E_2(i,j)| \geq p_3 \\ & or\ \ |E_3(i,j)| \geq p_5 \text{ or } |E_4(i,j)| \geq p_5 \\ 1, & p_0 < |E_1(i,j)| < p_1 \\ & or\ \ p_2 < |E_2(i,j)| < p_3 \\ & or\ \ p_4 < |E_3(i,j)| < p_5 \\ & or\ \ p_6 < |E_4(i,j)| < p_5 \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

where "2" denotes the presence of strong edge, "1" represents weaker edges, and "0" denotes no edge.
$\{p_0, p_1, p_2, p_3, p_4, p_5, p_6\}$ is a set of user-defined thresholds. The tone correction function f is modeled using the following equation

$$f(x) = \begin{cases} a^{\gamma-1}x, & 0 \leq x < a; \\ x^{\gamma}, & a \leq x < b; \\ (\frac{1-b^{\gamma}}{1-b})x + (\frac{b^{\gamma}-b}{1-b}) & b \leq x < 255. \end{cases} \quad (10)$$

The classification parameter set P can then be described as

$$P = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6, a, \gamma, b\} \quad (11)$$

Parameter Optimization

To optimally select the set P, a set of images scanned at 600dpi were used as a training set. Each image was segmented manually to produce the optimal segmentation $E_R(i,j)$. A cost function is then constructed to minimize the error in classification between the test and reference edge maps.

$$C = \frac{1}{MNK} \sum_{k=1}^{K} \sum_{i=1}^{M} \sum_{j=1}^{N} [E^k(i,j) - E_R^k(i,j)]^2, \quad (12)$$

where M and N are the image dimensions, and K is the number of images in the training set.

Genetic Algorithms

Genetic algorithms (GAs) belong to the class of stochastic search methods. Whereas most stochastic search methods work on a single solution at a time, GAs manipulates a population of solutions. GAs carry out simulated evolution on binary encoded solutions called chromosomes, which can be further concatenated into genes, to form the population.

When chromosomes/genes evolve, a new generation of solutions is produced. During the evolution and production process, the selection of the chromosomes is biased so that those with the best evolutions tend to reproduce more often than those with bad evolutions.

An evolution function is needed to describe the evolution performance of each chromosome/gene. This objective function is the link between the genetic algorithm and the problem to be solved. An evolution function takes a chromosome as input and returns a value that describes the fitness of each chromosome/gene.

Reproduction of chromosomes/genes is performed by a most probable operation called crossover. This operation recombines the genetic material in two parent chromosomes to make two children. Crossover is an essential and most frequent operation in the reproduction model. In addition to crossover, mutation is another reproduction operation. Mutation can cause a limited change to the genetic material of parents in very rare situations.

With encoding, evaluation, selection, and reproduction operations, GAs provides a powerful method of searching. As global search method, GA has the advantage over the local search methods of less sensitivity to the initial parameter selection and less chance of being trapped to local minima [18].

Optimization Using Genetic Algorithms (GAs)

To apply the GA to our problem, we encoded the transformation parameters as genes. Each parameter is encoded by 16 bits. The genes are formed by concatenating the binary coded parameters. The crossover operation occurs at multi points along the gene with probability $p_c = 0.90$. A mutation rate of 0.005 is usually used.

Since GA maximizes a fitness function, we redefined the cost function described in Eq. 12 to be

$$F = \frac{MNK}{\sum_{k=1}^{K} \sum_{i=1}^{M} \sum_{j=1}^{N} [E^k(i,j) - E_R^k(i,j)]^2}, \quad (13)$$

Results

A training set of three 600 dpi images was used. The training images included different contents ranging from smooth to sharp transition areas. Each image was first manually classified to produce an ideal classification map. The segmentation parameters were randomly initialized within a specified range for each parameter. These ranges are shown in Table 1. A genetic algorithm, with parameters shown ib Table 2, is applied to the input classification maps and the corresponding ideal maps. As the genetic algorithm progresses, new segmentation parameters are produced such that the miss-classifications can be reduced. This enhancement can be indicated by a monotonically decreasing error. The evolution of the segmentation parameters with the value of the fitness and error at each iteration/generation are shown in Figure 1. The initial and final values of the segmentation parameters and fitness are shown in Table 3.

Using the optimized parameters, an optimized classification map is generated as shown in Figures 2-6 (bottom) while the reference classification map is shown in Figure 2- 6(middle). Sharp transition are represented by red, while weak and smooth transitions are represented by green and blue, respectively. The optimized maps show more consistency with the reference maps. Consequently, better enhancement would be achieved by applying the correct filter to enhance different image regions.

Parameter Ranges.

Parameter	Range
p_0	10-30
p_1	25-90
p_2	80-90
p_3	25-90
p_4	25-90
p_5	25-90
p_6	25-90
a	1-80
γ	0.5-2.9
b	240-254

Genetic Parameters.

Parameter	Value
No. of Parameters	10
No. of Generations	300
Population size	20
No. bits	16
Crossover rate	0.9
Mutation rate	0.005

Optimized Parameters.

Parameter	Initial value	Final value
p_0	25	17
p_1	84	43
p_2	81	87
p_3	58	55
p_4	47	82
p_5	76	55
p_6	47	38
a	76	15.7
γ	1.05	0.96
b	203	236.47
Fitness	3.7	4.88

References

[1] M. Ahmed, B. E. Cooper, and S. Love, "Document Image Segmentation Using a Two-stage Neural Network," *Proc. of SPIE* , Vol. 3962, pp. 25-33, January 2000.

[2] M. Unser, A. Aldroubi, and M. Eden, "Enlargement or Reduction of Digital Images with Minimum Loss of Information," *IEEE Trans. on Image Processing* , Vol. 4, No. 3, pp. 247-258, March 1995.

[3] Y. Y. Tang, C. Y. Suen, and M. Cheriet, "Document Analysis and Understanding: A Brief Survey," *Proc. Int. Conf. on Document Analysis and Recognition*, pp. 17-31, 1991.

[4] H. Cheng and C. Bouman, "Trainable Context Model for Multiscale Segmentation," *Proc. Int. Conf. on Image Processing (ICIP)*, October 1998.

[5] L. O'Gorman, "The Document Spectrum for Page Layout Analysis," *IEEE Trans. on Pattern Matching and Machine Intelligence (PAMI)*, Vol. 15, No. 11, November 1993.

[6] K. Etemad, D. Doermann, and R. Chellapa, "Page segmentation using decision integration and wavelet packets," *Proc. Int. Conf. of Pattern Recognition*, Vol. 2, pp. 345-349, 1994.

[7] A.K. Jain and Y. Zhong, "Page segmentation using texture analysis," *Pattern Recognition*, Vol. 29, No. 5, pp. 343-770, 1996.

[8] H. Cheng, C. Bouman, and J. Allebach, "Multiscale Document Segmentation," *Proc. of IS&T 50th Annual Conference*, pp. 417-425, May 1997.

[9] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Image Segmentation Using Expectation- Maximization and its Application to Image Query," *IEEE Trans. on Pattern Matching and Machine Intelligence (PAMI)* , Vol. 24, No. 8, pp. 1026-1038, August 2002.

[10] A. P. Witkin, "Scale space filtering," *Proc. International Joint Conference on Artificial Intelligence*, pp. 1019-1023, 1983.

[11] S. Haring, M. Viergever, and J. Kok "A multiscale approach to image segmentation using Kohonen networks," *Proc. IPMI*, pp. 212-224, Berlin, 1993.

[12] J. J. Koenderink, "The structure of images," *Biological Cybernetics*, Vol. 50, pp. 363-370, 1984.

[13] L. Florack and B. M. Romeny, "Scale and the differential structure of images," *Image and Vision Computing*, Vol. 10, pp. 376-388, 1992.

[14] R. M. Haralik and L. G. Shapiro, "Computer and Robot Vision," Addison-Wesley Publishing Company, 1992.

[15] S. Dellepiane et al., "Model generation and model matching of real images by fuzzy approach," *Pattern Recognition*, Vol. 25, No. 2, pp. 115-137, 1992.

[16] S. Haykin, "Neural Networks: A Comprehensive Foundation," Macmillan College Publishing Company, 1994.

[17] T. Kohonen, "Self Organization and Associative Memory," Springer-Verlag, 1984.

[18] D. E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," Addison-Wesley Publishing Company, 1989.

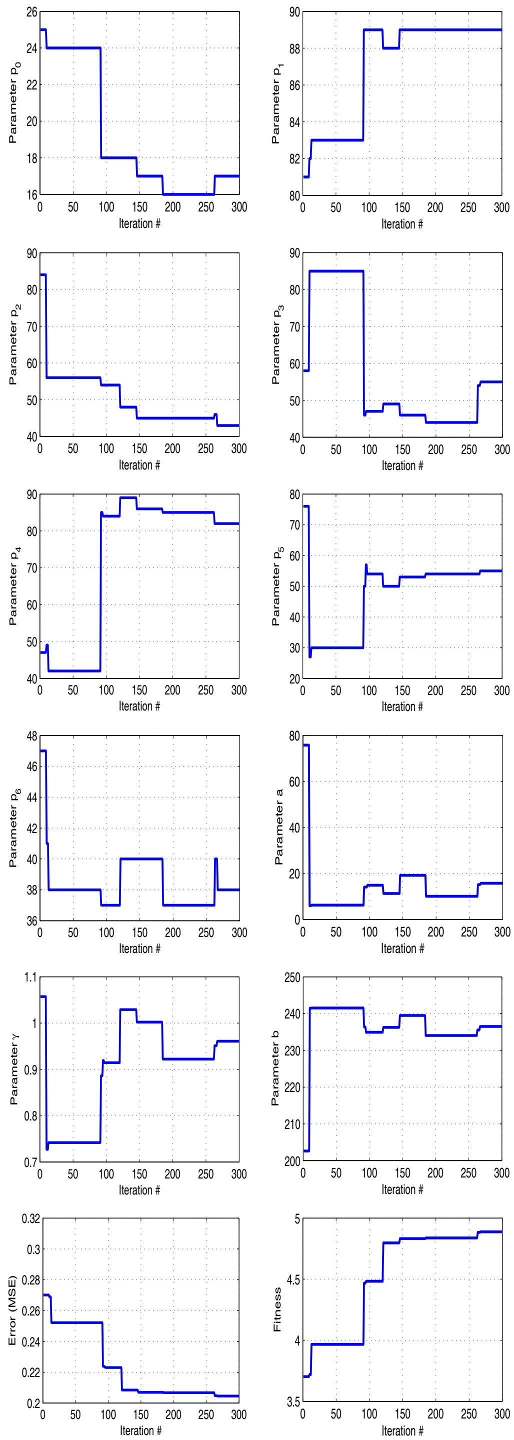

Figure 1. *Parameters optimization: A set of ten segmentation parameters and tone-correction curve are optimized using a genetic algorithm of population size 20; number of generations 300; crossover rate 0.9 and mutation rate of 0.005. The parameter values are shown at each iteration with the value of the error, as an indication of the number of classification, is monotonicaly non-increasing.*

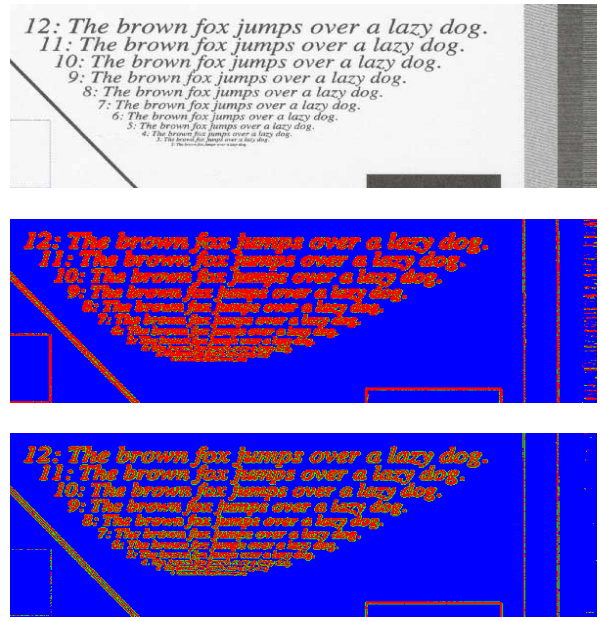

Figure 2. *Optimized classification: (Top) Orginal test image, (middle) the ideal classification and (bottom) the optimized classification.*

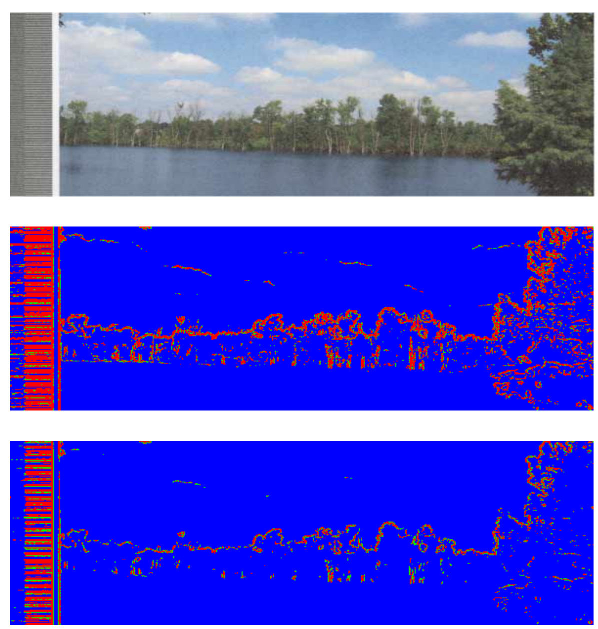

Figure 3. *Optimized classification: (Top) Orginal test image, (middle) the ideal classification and (bottom) the optimized classification.*

Figure 4. *Optimized classification: (Top) Orginal test image, (middle) the ideal classification and (bottom) the optimized classification.*

Figure 5. *Optimized classification: (Top) Orginal test image, (middle) the ideal classification and (bottom) the optimized classification.*

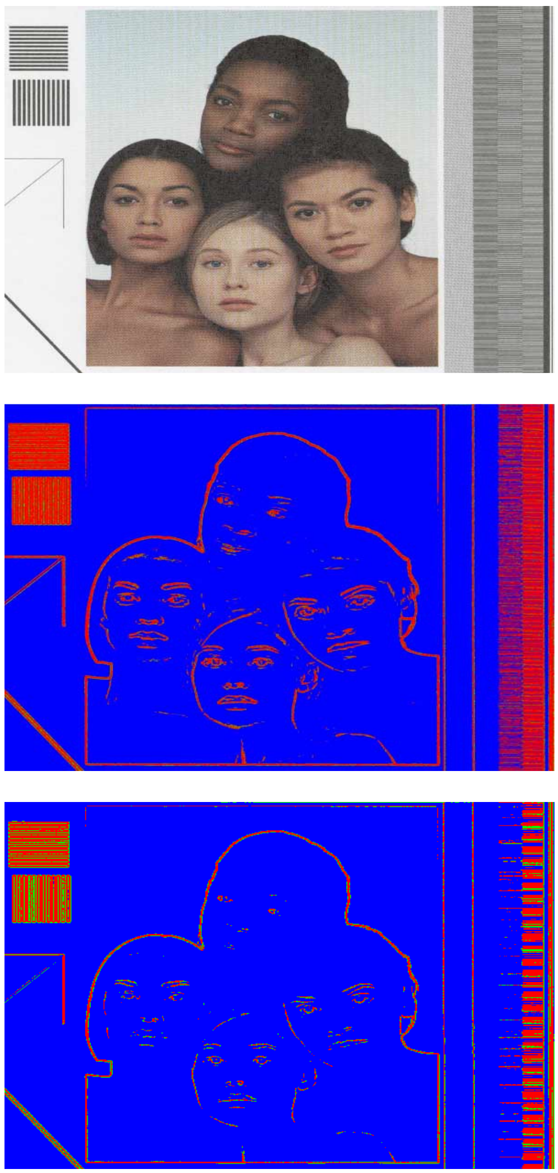

Figure 6. *Optimized classification: (Top) Orginal test image, (middle) the ideal classification and (bottom) the optimized classification.*