ELSEVIER

Pattern Recognition Letters 22 (2001) 1503–1515

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Neural network-based text location in color images

Keechul Jung *

Computer Graphics Lab., School of Elect. and Comp. Eng., Sung Kyun Kwan University, Chunchun-dong, Jangan-gu, Suwon, Kyunggi-do 440-746, South Korea

Received 26 June 2000; received in revised form 26 March 2001

Abstract

This paper proposes neural network-based text locations in complex color images. Texture information extracted on several color bands using neural networks is combined and corresponding text location algorithms are then developed. Text extraction filters can be automatically constructed using neural networks. Comparisons with other text location methods are presented; indicating that the proposed system has a better accuracy. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Text location; Neural network; Texture discrimination; Arbitration on color bands; Video indexing

1. Introduction

Recently, researchers have attempted to use the text-based retrieval of image and video data using several image processing techniques. An automatic text location algorithm for image data and video documents is very important as a preprocessing stage for optical character recognition (OCR). Accordingly, several approaches to text location in images have been proposed for specific applications including page segmentation, address block location, license plate location, and indexing for video archives (Jain and Chen, 1994; Jain and Karu, 1996; Jung et al., 2000; Jung et al., 1999; Kim et al., 1998; Park et al., 1999; Strouthopoulos and Papamarkos, 1998; Tan and Ng, 1998; Jain and Yu, 1998). In the extraction of text printed against a shaded or textured background or embedded in a complex image, there are many sources of variability in text region location. Text variations in terms of character font size, style, orientation, alignment, texture, and color embedded in low contrast and complex background images make the problem of automatic text location extremely difficult.

There are two primary methods for text location. The first method uses a connected component analysis method (Kim et al., 1998; Jain and Yu, 1998). This method applies a bottom-up approach by grouping small components into successively larger components until all blocks are identified on the image. However, the connected component method is not appropriate for video documents because it is based on the effectiveness of the segmentation method which guarantees that a character is segmented as one connected component separated from other objects. Accordingly, it is generally difficult to apply such a segmentation method to low-resolution video images with various noises. The second method regards text

* Tel.: +82-331-290-7223; fax: +82-331-290-7211.
E-mail address: kjung@ece.skku.ac.kr (K. Jung).

PII: S0167-8655(01)00096-4

regions as textured objects and applies gabor filters, wavelet decomposition, and spatial variance to exploit the textural properties (Jain and Chen, 1994; Jain and Karu, 1996; Jung et al., 1999; Patel, 1996; Strouthopoulos and Papamarkos, 1998; Randen and Husoy, 1999). This utilization of texture information for text location is also sensitive to character font size and style. So, in a complex situation with various font styles, sizes, and colors, it is difficult to manually generate a texture filter set for each application.

This paper uses a neural network-based texture discrimination method for text locations in color images and demonstrates that the proposed method is applicable to text locations in complex color images for efficient content-based indexing. Neural networks are employed to train a set of texture discrimination masks that minimize classification error for the given texture classes: text region and non-text region. Texture discriminations are performed by convolving the trained masks with the input image. This paper considers the textural properties of text regions on several color bands, that is, the texture information on several color bands is combined and corresponding object location algorithms are then developed. Different features can have different similarity measurements. This means that various features may play varying degrees of importance in making the final decision (Raghu et al., 1995). A neural network-based arbitration method is used to describe the influence of each color band on texture discrimination. For text locations, three neural networks that can separate text class from non-text class are constructed. The segmentation works by applying all the neural networks to an input image and arbitrating the output of each neural network. An arbitration neural network can combine the outputs of several detectors into a single decision about the presence of texts. The detection and arbitration neural networks are both trained using a supervised learning method to minimize any classification error. Comparisons with neural network-based texture discrimination using a gray-scale image and connected component-based one are presented; indicating that the proposed system has a better accuracy.

The remainder of the paper is organized as follows. Section 2 describes and analyses the proposed text location method along with an arbitration technique for multi-layer perceptrons. Experimental results and evaluations are shown in Section 3. Section 4 presents some final conclusions and outlines future work.

2. Neural network-based text location

2.1. Texture properties of text regions

This section discusses the textural properties of text regions in images. Most texture-based methods assume that an image has the desired textural characteristics. However in realistic applications (for example, text region location and face detection), one has to be sure whether the objects (texts, face, etc.) in the image have texture. Accordingly, before we continue this research, the textural properties of text regions in video documents are checked. Fig. 1 shows the textural properties of text regions in video documents on red color bands (Karu et al., 1996). Fourier spectra are displayed as intensity images. The horizontal and vertical axes represent the frequencies in each direction, respectively. Fig. 1(a) shows that the text region has its special orientation and localized frequency comparing with non-text images Fig. 1(b). We have a similar frequency response on green, blue, and intensity color band.

Fig. 2 shows an example of the cross-section of an image including the text regions. The cross-section, which is marked with a black horizontal line, is presented as red, green, blue, hue, saturation, and intensity values. Each color value is normalized to $[0, 1]$. The regular variations of the red, green, blue, and intensity levels represent the textural properties of the text regions, however, no specific properties can be detected in the hue and saturation bands.

2.2. Overview of algorithm

The proposed text location system operates in three stages: First, it applies neural network-based filters to the input image, next, it arbitrates

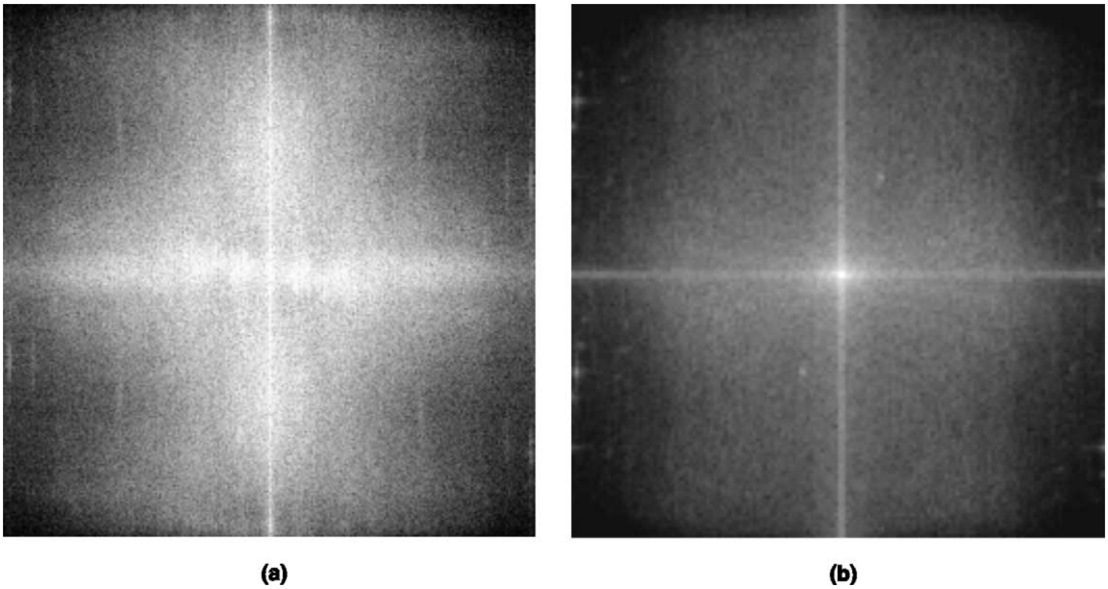

Fig. 1. Frequency responses of text regions in red color band: (a) for text regions and (b) for non-text regions.

between several outputs and finally, it post-processes (eliminates noises, then places bounding boxes). In the proposed method, neural networks are used to classify the pixels of input images, that is to say, feature extraction and the pattern recognition stage are integrated in the neural network. To classify the pixels in the input, filters are convolved at each location in the image. That is, the neural networks examine local regions looking for text pixels that may be contained in a text region. They receive an $M \times M$ pixel region of the image as the input and generate a classified image as output. A neural network-based arbitration method is used to describe the influence of each color band. An arbitration neural network produces a final result based on the outputs of the three neural networks. After the pattern passes the network, the value of the output node is compared with a threshold value and the class of each pixel is determined. As a result of this classification, a classified image is obtained. During the post-processing stage, the classified image is smoothed so that any sparsely distributed text pixels are eliminated from the text class whereas any non-text pixels near densely distributed text pixels are included into the text class. Rectangles surrounding the text regions are then located by projected profile processing. Located rectangles that are too narrow to contain textual information are rejected. Furthermore, two rectangles in the same line are merged if they are very close on the x-axis. Fig. 3 shows the structure of the network for text location.

2.3. Neural network-based filters and arbitration

Two neural network structures are used: one performs texture analysis for each color band; the other arbitrates between each individual neural network. The filtering algorithm works by applying neural networks directly to the sub-regions of the input image. An input image is segmented into text and non-text classes using multi-layer feedforward neural network classifiers which receive the color values of a given pixel and its neighbors as input. The activation values of the output node are used to determine the class of a given central pixel. To further reduce the number of false positives, multiple networks can be applied and their outputs arbitrate to produce the final decision. Three methods are used to combine the outputs of three neural networks: ANDing, ORing, and the

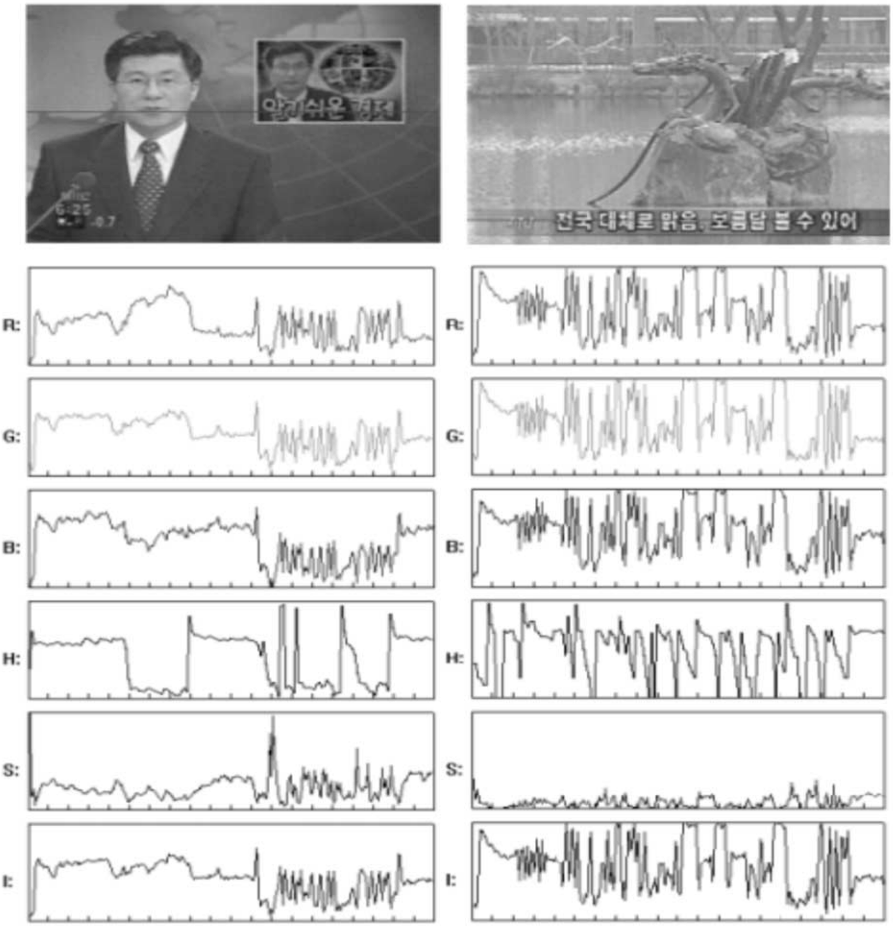

Fig. 2. Color variation examples of horizontal lines of text regions in images.

use of an arbitration neural network. As a result of this arbitration, a classified image is obtained as a binary image in which the text pixels are black. The following describes the architecture of the neural network-based classifier. The filtering neural network performs similar operations with the multi-channel filtering methodology because of their similarities in structures (Jain and Karu, 1996). Adjacent layers are fully connected, the hidden layer operates as a feature extraction module, and the output layer is used to determine the class of a pixel: text or non-text. A schematic diagram of the neural network-based classifier is shown in Fig. 4. The input layer receives the color values of the pixels, at predefined positions inside an $M \times M$ window over an input frame. The network parameters that can be varied are the number of layers, number of nodes in each layer, and the mask size. The size of the input window is unfixed; therefore, the window size can be selected during the experiment in order to use the texture information as clearly as possible. The experiments are conducted using various input window sizes 3×3, 5×5, 7×7, 9×9, 11×11, 13×13, 15×15, and 19×19, the number of nodes in each hidden layer is set to 30, and 50, and the feature sets used are red, green, blue, hue, saturation, and intensity bands.

The training of a texture classifier is a two-stage process. Every neural network for each color band

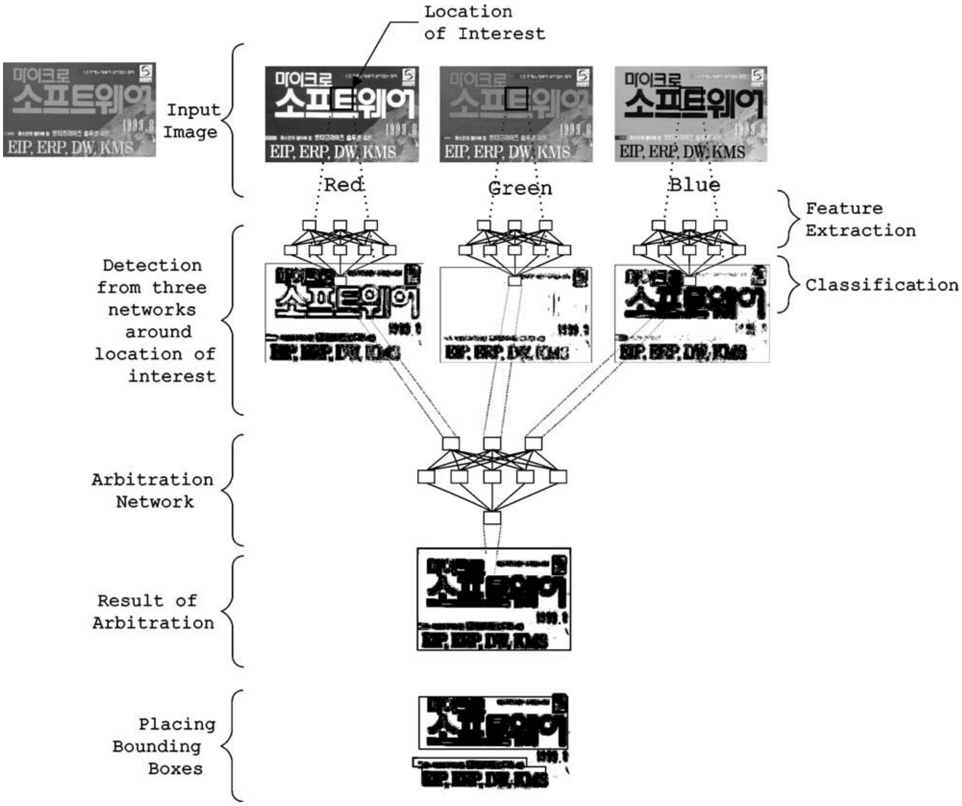

Fig. 3. Architecture of discrimination network that can arbitrate among multiple text detection networks.

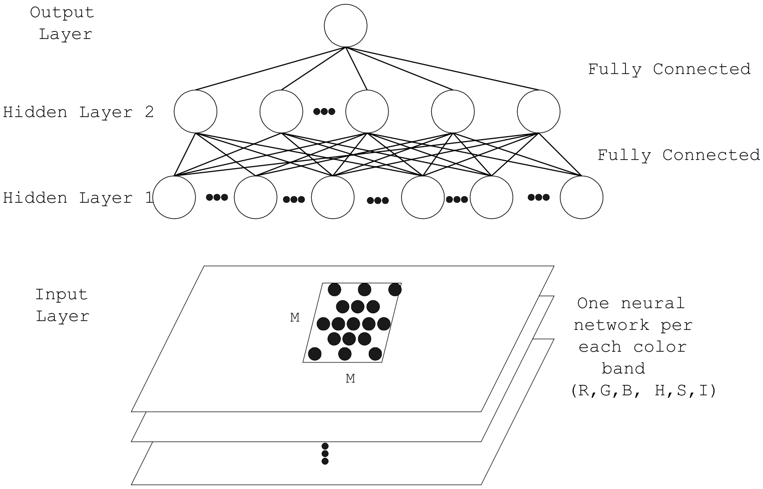

Fig. 4. Three-layer feed-forward neural network (Jain and Karu, 1996).

is initially trained with raw input images using a back-propagation learning algorithm. Thereafter, the last arbitration neural network is trained with the output of the previous three networks using the same back-propagation algorithm. The network is trained with a supervised learning method to minimize the classification errors. In several training protocols such as on-line, batch, and stochastic trainings, we use the batch mode. So, we present all the training patterns to the network before learning takes place. We use a constant learning rate $\eta = 0.02$, momentum $m = 0.5$ and the sigmoidal activation function of the form $f(\mathrm{net}) = a\tanh(b\,\mathrm{net})$, with $a = 1.7$ and $b = 0.75$. During the training session, a set of training patterns is used to train the weights of the network. Each training pattern consists of the color values of a pixel and its neighbors, along with the actual class of the pixel. If the value for classification is larger than the threshold value, it is considered as text. As the threshold value is changed in the training and testing stage, the conservatism of the system can be varied. To select a proper threshold value, we test the system performance varying the threshold value in [0.3–0.7]. In this experiment we can achieve almost the same results in [0.4–0.7] as shown in Table 1. Accordingly, the threshold value is set at 0.5.

A comparison of the learning rates for the six types of neural networks is shown in Fig. 5. Horizontal and vertical axes denote iteration and error rates, respectively. It can be seen that the neural networks using red, green, blue, and intensity feature could learn this problem, however, it is not adequate to use a neural network using hue or saturation feature. We use the number of the iteration epoch as a stopping criterion. It can be seen that 2000 epochs are sufficient for convergence and no substantial decrease is expected in these several cases. For the convenience of reporting errors, the actual class of every pixel in each image in this study's database is manually labeled by marking the coordinates of all the text rectangles. The classification errors are then automatically computed by comparing the network's output with the actual labeled class corresponding to each pixel. The value of the classification error is the proportion of falsely discriminated pixels relative to the total number of pixels. For the non-text training data, the *bootstrap method* is used (Sung, 1996). Collecting non-text training images is difficult. Practically any image can be used as non-text samples. Some examples of non-text samples are collected during training. Plus, the partially trained system is applied during training to images which does not contain text.

Table 1
Recognition rates according to the threshold value

Threshold value	Recognition rates (%)
0.3	63.2
0.4	79.2
0.5	82.0
0.6	81.7
0.7	79.2

The various ways of integrating three network outputs are by ANDing, ORing, and Neural network-based Arbitrating. ANDing considers a detection if all outputs from 3 networks detect a text region, ORing considers a detection whenever any network detects text region, and Neural network-based Arbitration is trained to produce a positive output for a given set of inputs only if that location contains a text.

2.4. Analysis of networks

This section outlines the properties of the filtering neural networks. To do this, the frequency responses of the hidden nodes, the performance of the neural network according to the network configurations (window size and number of hidden nodes), and the effect of arbitration methods are all shown.

To investigate the properties of the masks, the frequency response of these masks must be demonstrated. To illustrate the frequency responses of the hidden nodes, the image shown in Fig. 6(a) was considered (Jain and Karu, 1996). Figs. 6(b)–(g) are the outputs of the first hidden layer's nodes when applied to Fig. 6(a). These figures were extracted from the neural network using intensity values as the input feature. For good visualization, the output images were smoothed and the contrast of the images enhanced. As we can see the local-

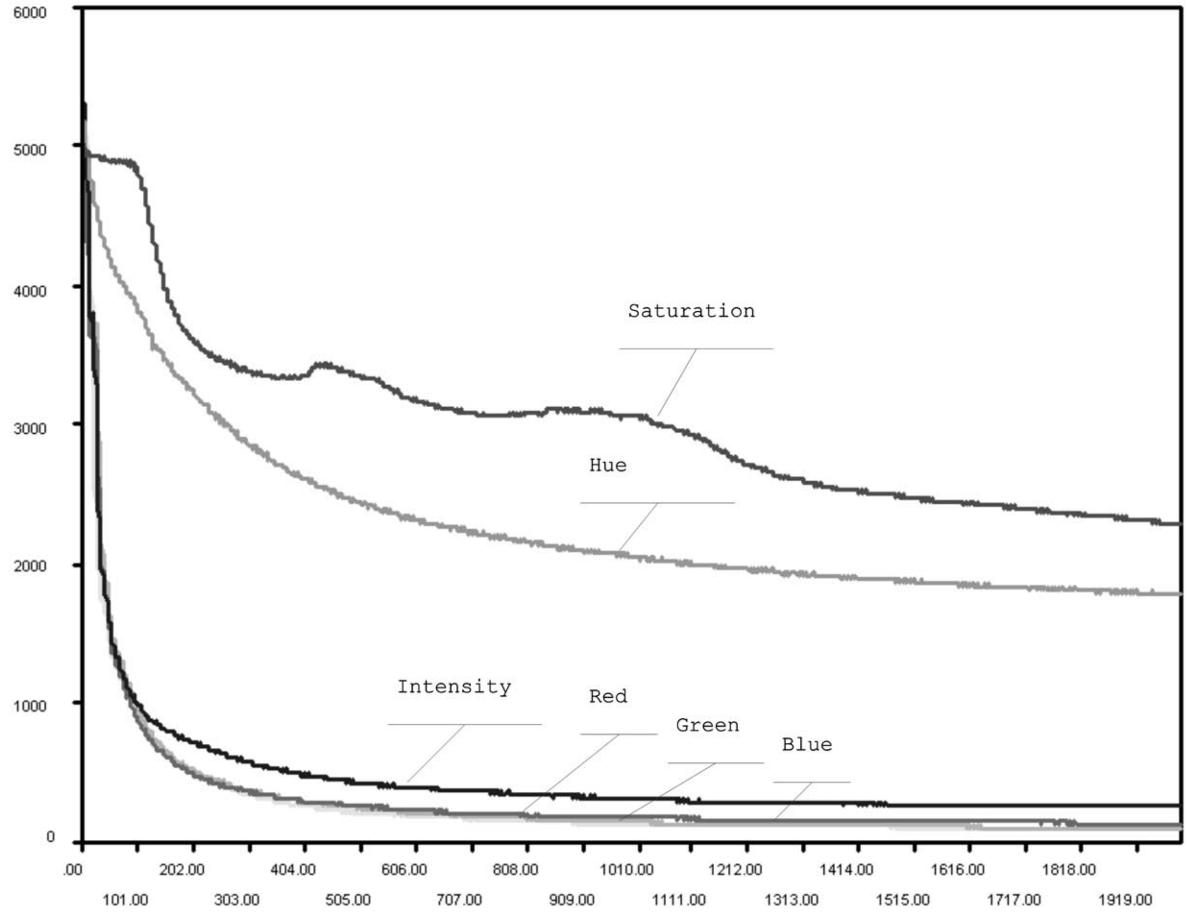

Fig. 5. Behavior of the neural network: convergence of training errors according to feature sets used.

ized response to the sample image, we can consider that each hidden node has its special orientation and localized frequency and performs similar actions with filters in multi-channel filtering in texture classifications.

The use of smaller size input windows can reduce the processing time. However, when a small size window is used, classification error increases. Choosing an appropriate input window size is a tradeoff between classification accuracy and processing time. The experiments were performed with input window sizes of 3 × 3, 5 × 5, 7 × 7, 9 × 9, 11 × 11, 13 × 13, 15 × 15 and 19 × 19. This experiment also used a two hidden layers structure with 30 nodes per hidden layer and a neural network with intensity values as the input.

As summarized in Table 2, the mis-classification rates according to the input window size were 24.8%, 22.8%, 23.5%, 18.8%, 21%, 18%, 20.9%, and 24.7% respectively. However, when the classification results were smoothed, they became 21.7%, 19.8%, 20.4%, 15.2%, 18.8%, 13.7%, 15.7%, and 21.2%, respectively. The size selection of the moving window depends on the text size. Increasing the input window size does not always decrease the error rates because of the boundary approximations. Similar results for texture discrimination are reported in reference (Jain and Karu, 1996) and (Bhattacharya et al., 1997). The rather low segmentation rate for the classifier that used 15 × 15 and 19 × 19 input windows could be attributed to the unstable classification of the

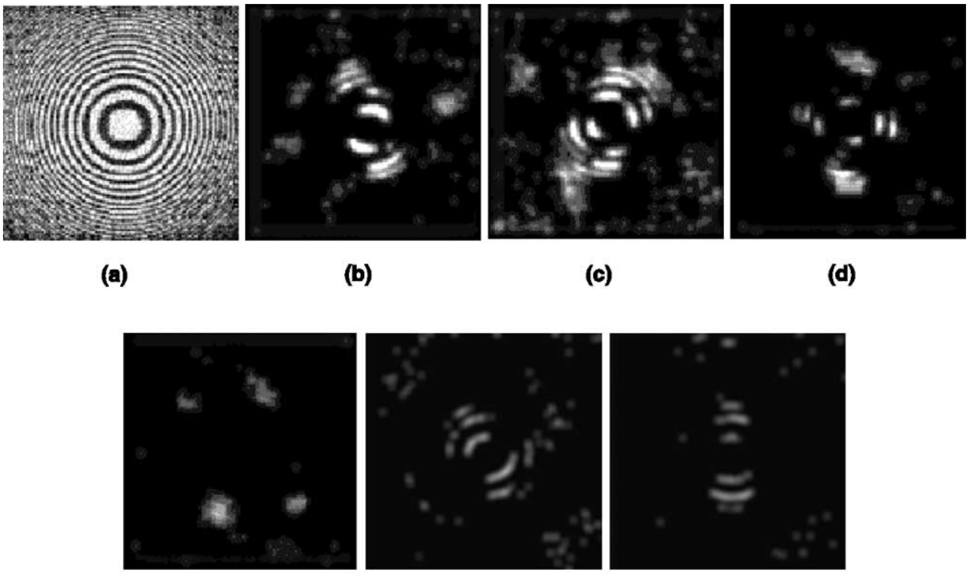

Fig. 6. Sample outputs ((b)–(g)) of hidden nodes when applied to 128 × 128 image in (a).

Table 2
Mis-classification rates relative to input window size

Window size	Using texture classifier alone (%)	With smoothing (%)
3 × 3	24.8	21.7
5 × 5	22.8	19.8
7 × 7	23.5	20.4
9 × 9	18.8	15.2
11 × 11	21	18.8
13 × 13	18	13.7
15 × 15	20.9	15.7
19 × 19	24.7	21.2

patterns owing to the neural network's generalization performance and lack of training data comparing with the size of neural network. The system performance depends on the text size and input window size. When we use the 13 × 13 size window for input window, we can get a best performance in the 10–19 size characters (Table 8). Fig. 7(a) presents the test input images, and Figs. 7(b) and (c) show the results of image classification using 3 × 3 and 13 × 13 input windows. The result from the 3 × 3 window size filtering network resembles the outputs of the edge filters

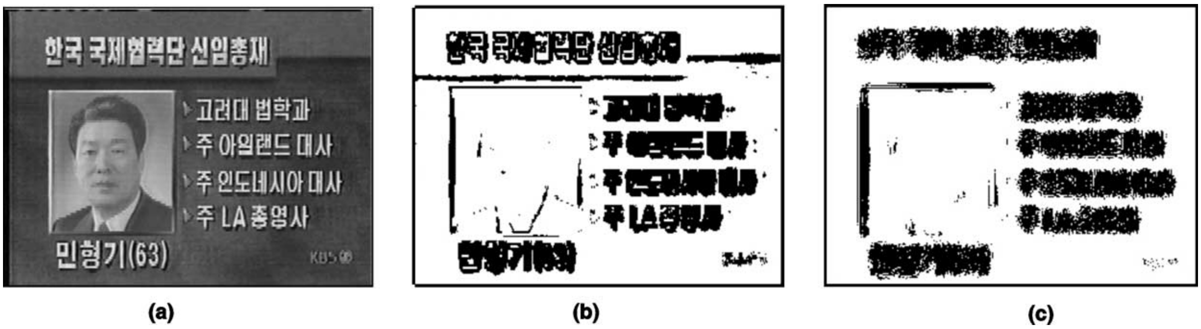

Fig. 7. Experimental results: (a) test images, (b) 3 × 3 window, and (c) 11 × 11 window.

and includes many errors. Table 3 shows the performance of different versions of the detector on the test images. This table shows the total misclassification rates after noise elimination. Table 3 shows the results for 12 individual neural networks. The first used 30 hidden units per hidden layer. The second used 50 hidden nodes, with complete connections between each layer. The input window size was 13. The classification rates of red, green, blue, and intensity were nearly the same as each other.

The use of arbitration further reduces the false detection rate. Note that, for systems using neural network-based arbitration, the ratio of mis-classification is extremely low. Whereas, the outputs from the ANDing operation only show slightly lower 'Mis-classification Rates' than ORing, only slightly higher 'Missing Rates' than ORing. Here missing rate means the ratio of recognizing text pixels as non-text ones, and the mis-classification does the ratio of classifying text pixels as non-text pixels and including opposite case (Table 4).

3. Experimental results

3.1. Data preparation

A number of experiments were performed to evaluate the system. The proposed text location method was applied to 12 video clips. Each video clip had a running time of 2–3 min, and included broadcast news from the Munhwa Broadcasting Corporation (MBC) and Korean Broadcasting System (KBS), plus sample files from web site of MoCA Project (Lienhart and Stuberm, 1996). 1000 key frames with a size of 320×240 were automatically selected using a simple key frame extraction technique. Out of these frames, 50 video frames were used in the initial training process, and the others were used in the testing process. This paper focuses on super-imposed and horizontally aligned text in video frames. No prior knowledge of resolution, text location, and font styles is assumed. However, size restrictions (characters cannot be too small to be read by humans or too big to occupy a large portion of the

Table 3
Mis-classification rates according to network configuration

System	Feature type	Mis-classification rates (%)
Network 1 (30 hidden nodes per hidden layer)	Red	13
	Green	15
	Blue	12
	Hue	87.2
	Saturation	54.8
	Intensity	13.7
Network 2 (50 hidden nodes per hidden layer)	Red	12.2
	Green	14.3
	Blue	13.3
	Hue	89.7
	Saturation	54.3
	Intensity	13.2

Table 4
Detection rates according to arbitration method

Type	Neural network size	Missing rate (%)	Mis-classification rate (%)
Neural network-based arbitration between R, G, B networks	30 Hidden nodes	30.2	7.8
ANDing between R, G, B networks	30 Hidden nodes	45.3	13.4
ORing between R, G, B networks	30 Hidden nodes	27	12

frame) are required. The size of the text in the video frames ranged from 12 × 13 to 21 × 23. In many researches about the color texture analysis, usually La^*b^* appears to provide a better performance than RGB for image processing tasks such as color texture segmentation, However the RGB shows better performance in terms of noise-sensitivity (Pascho and Valavanis, 1999). The video frames which are used for this experiment, captured in the size of 320 × 240, have many salt and pepper noise and they are degraded because of MPEG encoding. So, we use the RGB color space.

3.2. Post-processing

Some pixels can be determined as belonging to a different class from their actual class. Noise elimination is accomplished on an output image from the arbitration neural network using median filter. Actually the output image from the neural network filter has many salt and pepper noise. And we have to preserve the border of the text regions for the bounding box stage. So we use the median filters.

During this step, text rectangles are identified by performing a profile analysis and merging certain rectangles. We only concentrate on horizontally aligned texts in input images, so, we can use simple heuristic method to align bounding boxes. A smoothed image is projected along the y-axis and a profile is computed. A *text zone* is defined as a consecutive vertical zone in which the profile values are more than a threshold. Each text zone is also projected along the x-axis and another profile is computed. A *text segment* is then defined as a consecutive horizontal zone in which the profile values are more than a threshold. In this experiment, the threshold values for the x-axis and y-axis profiles were selected as half and two-thirds of the highest value in the profiles, respectively. In addition, the following three heuristics are used to remove text segments that hardly included any characters: (1) The height of a text segment should be larger than 1/30 of the height of the image;

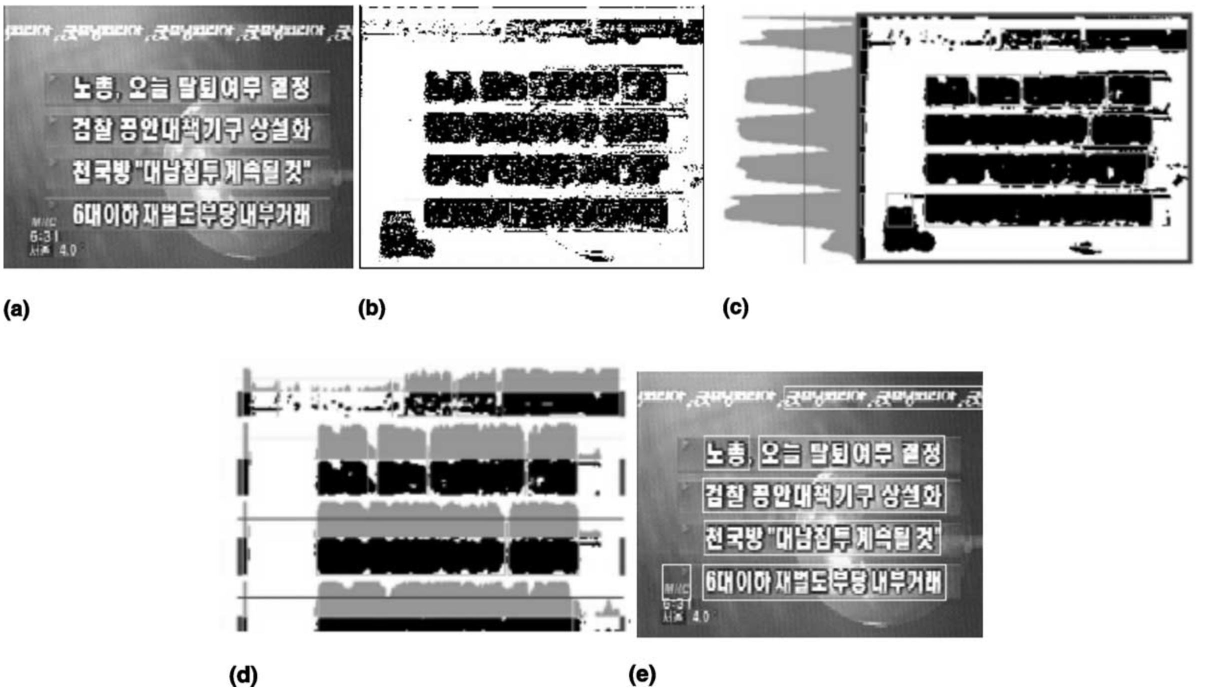

Fig. 8. Example of post-processing: (a) input image, (b) smoothed image, (c) horizontal projection profile, (d) text zone using vertical projection profile, and (e) final result.

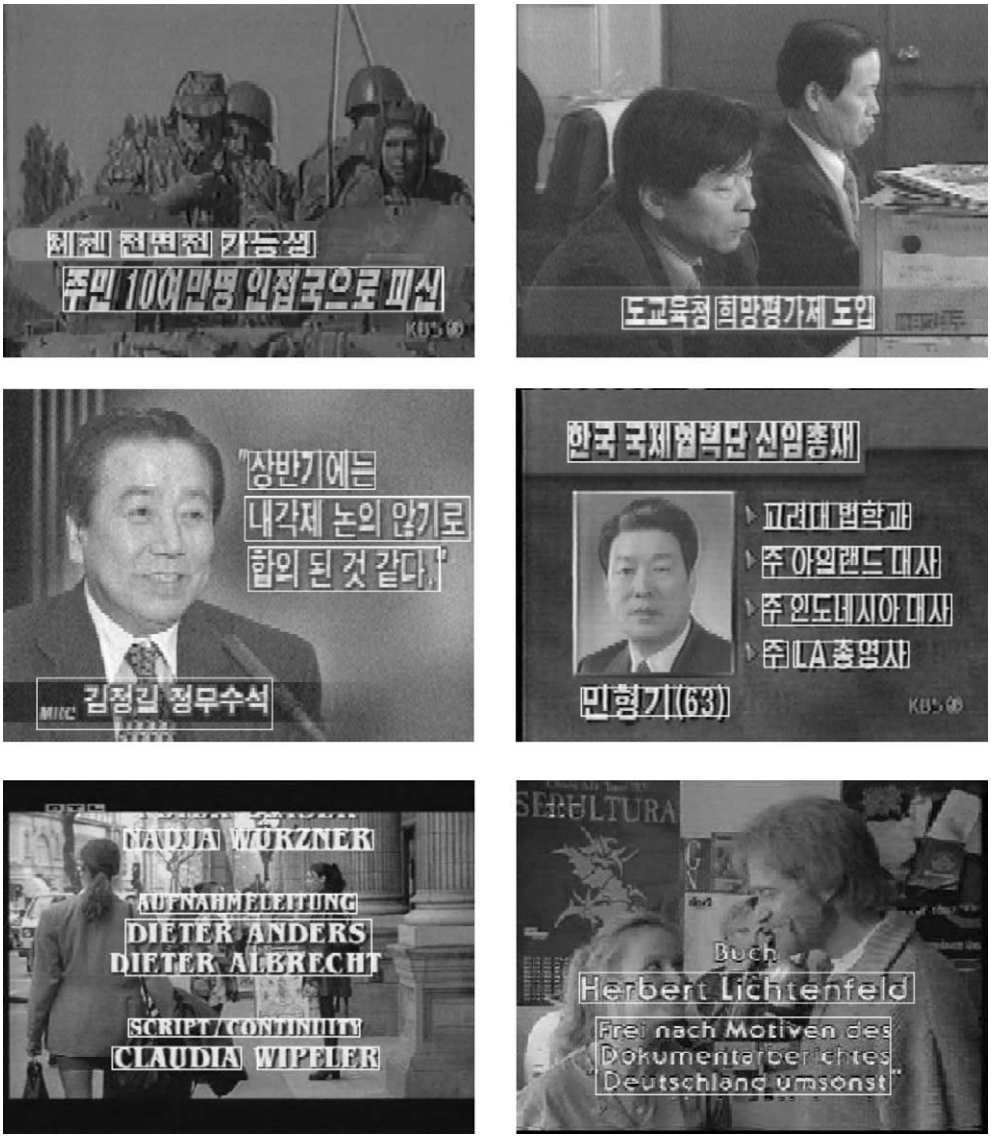

Fig. 9. Example of text location.

Table 5
Processing times for several routines

	Filtering using neural network	Noise elimination	Bounding box
Processing time (s)	3.5	0.4	0.1

(2) the width of a text segment should exceed half of the height; (3) two text segments in the same text zone should be distant from each other by more than 1/5 of the height. Accordingly, any segments that violate (1) and (2) are no longer considered as text segments. Plus any two text segments violating (3) are merged into one segment. *Text rectangles* are then obtained by merg-

Table 6
Experimental results

		Arbitration with NN	Using only intensity as the input
Mis-classification rates (%)	Before placing bounding boxes	7.8	13.7
	After placing bounding boxes	6.8	12.2

Table 7
Comparison of classification rates using connected component method

	Extraction rates	Processing time (s)
Proposed method	92.2	11.3
Connected component method (CCM)	73.8	1.2
Modified CCM	82.1	4.6

ing nearby text segments after rejecting the segments violating the above rules. Figs. 8(c)–(d) illustrate how text zones and text segments are extracted. Fig. 8(c) is a y-axis projection profile of a smoothed image, and Fig. 8(d) is an x-axis projection profile of an extracted text zone. Fig. 8(e) shows the final result of text location. The white rectangle in the final result represents a text rectangle.

3.3. Results of text locations

As mentioned earlier, several experiments were performed to locate texts in video frames. The proposed algorithm requires that several parameters are set empirically, which can then be tuned for a particular class of image. Accordingly, the threshold value for the output of the discrimination was set at 0.5, the two hidden-layer perceptrons included 30 hidden nodes per layer, and the input window size was 13. Three neural networks (for red, green, and blue) were used for the texture discrimination and neural network-based arbitration. Fig. 9 shows examples of the text location.

3.4. Evaluation

First the processing time for one neural network was tested, as shown in Table 5. Table 6 shows a comparison with the case of using only intensity as the input feature. The proposed method located 92.2% of the text regions and 93.2% after placing bounding boxes. It shows a better performance comparing with the case of intensity feature only.

Table 8
Experimental results according to text size

Text size (pixels)	Detection rate (%)
~9	–
10–19	92.3
20–29	89.3
30–39	–

Table 7 shows a comparison with the connected component analysis method (CCM) used in references (Zhong et al., 1995) and modified CCM (Kim and Jung, 2000), and it is clear that the proposed method exhibits a superior performance than the connected component methods. The connected component methods were applied after converting input images into the gray-scale images. We quantize the color space into a few prototypes using the method found in (Zhong et al., 1995). We use these several local maxima in a color histogram of the input image as prototypes. So, we could get a quantized image using these prototype colors. Table 8 shows the detection rates relative to the text sizes. In the case of using a 13 × 13 input window size, text sizes from 10 to 19 exhibited the best performance.

4. Conclusions

This section presents a summary of the text location method proposed in this paper. Plus some continuing problems are noted, which will need to be addressed in future work. It has been shown that neural networks can be trained for text locations in images. The frequency responses of the hidden nodes, which show that neural networks have a gabor filter-like localized frequency and orientation selectivity, let us exploit the neural network as a text detector. Text filters for several text styles can be automatically constructed using neural networks. And a superior performance can be achieved for complex images through the use of several color bands instead of gray-level input features.

There are a number of directions for future work. The main limitation of the current system is its running time. The dominant factor in the running time of the proposed system is the number of convolution with neural network-based filter. Because convolving images with some filters are computationally expensive, we now try to find an efficient detector eluding the full scan of input images. And we also try to exploit temporal information on the digital video sequence.

References

Bhattacharya, U., Chaudhuri, B.B., Parui, S.K., 1997. An MLP-based texture segmentation method without selecting a feature set. Image Vision Comput. 15, 937–948.

Jain, A.K., Chen, Y., 1994. Address block location using color and texture analysis. Comput. Vision Graphics Image Process. 60 (2), 179–190.

Jain, A.K., Karu, K., 1996. Learning texture discrimination masks. IEEE Trans. Pattern Anal. Machine Intell. 18 (2), 195–205.

Jain, A.K., Yu, B., 1998. Automatic text location in images and video frames. Pattern Recognition 31 (12), 2055–2076.

Jung, K., Jeong, K.Y., Kim, E.Y., Kim, H.J., 1999. Neural network-based text location for news video indexing. In: Proc. Internat. Conf. on Image Processing.

Jung, K., Jeong, K.Y., Kim, K.I., Kim, H.J., 2000. Neural network-based OCR for video indexing. In: Internat. Conf. on Application and Pattern Recognition and Digital Techniques'99, pp. 321–325.

Karu, K., Jain, A.K., Bolle, R.M., 1996. Is there any texture in the image? Pattern Recognition 29 (9), 1437–1446.

Kim, E.Y., Jung, K., 2000. Automatic text region extraction using cluster-based templates. In: 4th Internat. Conf. on Advances in Pattern Recognition and Digital Techniques (ICAPRDT), pp. 418–421.

Kim, E.Y., Jung, K., Jeong, K.Y., Kim, H.J., 1998. OCR of Image on the WWW. In: Internat. Conf. on Electronics, Information, and Communications, Vol. 2, pp. 287–290.

Lienhart, R., Stuberm, F., 1996. Automatic text recognition in digital videos. SPIE – Internat. Soc. Opt. Eng., 180–188.

Park, S.H., Kim, K.I., Jung, K., Kim, H.J., 1999. Locating car license plates using neural networks. IEE Electron. Lett. 35 (17), 1475–1477.

Pascho, G., Valavanis, K.P., 1999. A color texture based visual monitoring system for automated surveillance. IEEE Trans. Syst., Man Cybernet. C 29 (1), 298–307.

Patel, D., 1996. Page segmentation for document image analysis using a neural network. Opt. Eng. 35 (7), 1854–1861.

Raghu, P.P., Poongodi, R., Yegnanarayana, B., 1995. A combined neural network approach for texture classification. Neural Networks 8 (6), 975–987.

Randen, T., Husoy, J.H., 1999. Filtering for texture classification: a comparative study. IEEE Trans. Pattern Anal. Machine Intell. 21 (4), 291–310.

Strouthopoulos, S., Papamarkos, N., 1998. Text identification for document image analysis using a neural network. Image Vision Comput. 16, 879–896.

Sung, K.-K., 1996. Learning and example selection for object and pattern detection. Ph.D. Thesis, MIT AI LAB.

Tan, C.L., Ng, P.O., 1998. Text extraction using pyramid. Pattern Recognition 31 (1), 63–72.

Zhong, Y., Karu, K., Jain, A.K., 1995. Locating text in complex color images. Pattern Recognition 28 (10), 1523–1535.