JOURNAL OF KOREA MULTIMEDIA SOCIETY, VOL. 7, NO. 12, DECEMBER 2004(pp. 1692-1699)

Face Detection Based on Thick Feature Edges and Neural Networks

Young Sook Lee†, Young Bong Kim††

ABSTRACT

Many researchers have developed various techniques for detection of human faces in ordinary still images. Face detection is the first imperative step of human face recognition systems. The two main problems of human face detection are how to cutoff the running time and how to reduce the number of false positives. In this paper, we present frontal and near-frontal face detection algorithm in still gray images using a thick edge image and neural network. We have devised a new filter that gets the thick edge image. Our overall scheme for face detection consists of two main phases. In the first phase we describe how to create the thick edge image using the filter and search for face candidates using a whole face detector. It is very helpful in removing plenty of windows with non-faces. The second phase verifies for detecting human faces using component-based eye detectors and the whole face detector. The experimental results show that our algorithm can reduce the running time and the number of false positives.

Keywords: Face detection, neural networks, edge image, computer vision, machine learning, pattern recognition

1. INTRODUCTION

Face detection is the first imperative step of human face recognition systems in computer vision, which is to localize and extract human face in still images or digital video sequences. It also has been developed for diverse applications such as tracking and surveillance system, video conferencing, video coding, content-based image and video retrieval, intelligent human-computer interfaces and so forth.

Over the past ten years, many researchers have introduced manifold features to neural networks, principle component analysis, support vector machines, independent component analysis or other methods for face detection[2,3,9-15]. Sung and Poggio[5] developed an example-based learning approach for locating vertical frontal views of human faces in complex scenes. This method made a model on the distribution of human face patterns by means of a few view-based "face" and "non-face" model clusters. At each location of an image that may include an human face, a difference feature vector is computed between its local image pattern and the distribution-based model. A trained classifier using difference feature vectors determines whether or not a human face exists at the given image location. Viola and Jones[4,17] proposed a very fast AdaBoost-based learning algorithm for frontal face detection, which employed a small number of critical visual features. They introduced a new image representation called the "Integral Image", which allows the features used by cascading classifiers. Yang et al [2] presented a SNoW-based face detector. The SNoW learning architecture is a sparse network of linear functions over the pre-defined or incrementally learned fea-

※ Corresponding Author : Young Sook Lee, Address : (608-737) 599-1, Daeyon-3 Dong, Nam-Gu, Busan, S. Korea, TEL : +82-51-620-6396, FAX : +82-51-620-6390, E-mail : yulisis@mail1.pknu.ac.kr
Receipt date : Sep. 18, 2004, Approval date : Dec. 13, 2004
† Dept. of Computer Science Graduate School, Pukyong Nat'l Univ.
†† Dept. of Electronics, Computer and Telecommunication Engineering, Pukyong Nat'l Univ.
(E-mail : ybkim@pknu.ac.kr)

ture space and is specially tailored for learning the presence of a very large number of features.

Many neural network based algorithms to localize and detect the human face have been widely utilized because of the accuracy of detection and simple and efficient implementation. However, they show very expensive running time. The most significant work is the algorithm given by Rowley et al[1]. This work applied one or more neural networks directly to several portions of the input image and arbitrating its results. Each network has been trained to output the presence or absence of a face. They presented a face detection system based on retinally connected neural networks. Although most of neural network based face detection algorithms can make the excellent detection results, they spend plenty of time on the verification of a small window containing a face in an input image.

Recently, many researchers have interested in the reduction of false alarm rates, high detection rates, and speedup of the performance. They also have focused on extracting faces in still images and digital video sequences. Most of these methods spent plenty of the running time on reducing false alarm rates and vice versa. It is very difficult to control trade-off between the reduction of false alarm rates and the speedup of running time.

Therefore, we propose a frontal and near-frontal face detection algorithm in still gray images using a thick edge image and neural network. We tried to improve the system performance by speeding up the running time and reducing false alarm rates.

The remainder of this paper is organized as follows. An overview of and the description of our algorithm described in Section 2 and Section 3, respectively. Experimental results in our system are given in Section 4. The conclusions and future works are discussed in Section 5.

2. SYSTEM OVERVIEW

An overview of our face detection algorithm is given in Fig. 1. Our system is composed of two main phases: In the first phase, we applies twice a new edge filter to an preprocessed input image in order to remove the areas that might not contain a face. As the result, we can get a thick edge image. Pyramid images from the thick edge image in phase one or an original image in phase two resizes each search window to 25×25 pixels. We assume that faces in thick edge images are composed of constant edge feature vectors. When the number of edge feature vectors in a resized edge image is over a constant threshold under experiment, it is considered as an input to a neural network detector. It can allow to discard lots of windows containing non-faces before input to the

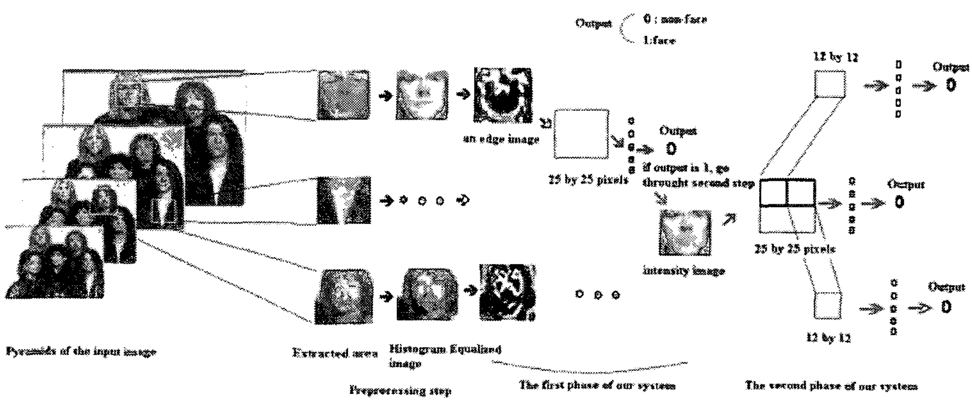


Fig. 1. Overview of our face detection system.

neural network detector. We also use the thick edge image in order to obtain the vivid edges that consist of boundaries of a face. At each level of image pyramid, the resized thick edge image is first passed through the neural network detector trained to detect a whole face. The system determines whether or not a small search window contains a face. When a search window is turned out to include a face, the 25×25 original intensity image in corresponding to the search window becomes an input for a neural network filter in the second phase. To obtain more accurate and robust face detection, we utilize one detector for detecting a 25×25 whole face and other component-based detectors considering the features of 12×12 left- and right-eye.

If we combine the outputs of three detectors, it will be very precise to select candidate windows containing a human face.

3. PROPOSED FACE DETECTION ALGORITHM

3.1 Feature Extraction: a Thick Edge Image

Many previous works have been performed a preprocessing step such as light correction and histogram equalization. The histogram equalization is to assign a uniform histogram to the input image. This equalization can correct different camera gains and improve the contrast. Thus, it makes the images independent on the environments. Illumination correction subtracts a best-fit brightness plane from a given resized window pixels. It can reduce the effect of heavy shadows caused by extreme lighting angles. However, we didn't apply to illumination correction in our algorithm. We only add the histogram equalization to the preprocessing step of our system. If we apply to illumination correction, our system will improve the performance more.

Initially, an original image goes through the preprocessing step using histogram equalization. The preprocessed image is inputted to a 3×3 devised edge filter on the given image at each pixel position and then resulted in a thick edge image. Image pyramids from an image is formed to detect different size of faces. Each level of the pyramid is given by applying a scale factor of 1.2. We utilize that pyramid images from the thick edge image at phase one or an original image at phase two resizes each search window to 25×25 pixels. In this work, we assume that faces in thick edge images consist of constant edge feature vectors. If the number of edge feature vectors in a resized edge image is within constant threshold ranges under experiment, it is considered as an input to a neural network detector.

We use the thick edge image in order to obtain the vivid edges that consist of boundaries of a face. At each level of image pyramid, the resized thick edge image is first passed through the neural network detector trained to detect a whole face. Using these thick edge images, we can eliminate lots of windows that not potentially contain a face. Thus, the speedup of our algorithm is greatly depend on the efficient generation of the thick edge image.

As one filter for obtaining thick edge image, we develop the new edge filter that combined following equation (1) and (2). In order to make this edge filter, we first make a new operator whose coefficients are different with the Prewitt operator[8] or the Sobel operator[8]. The Sobel operator and the Prewitt operator approximate the first derivative of the image in a given direction. The Sobel operator is sensitive to the diagonal direction and the Prewitt operator is sensitive to the horizontal or the vertical direction. Thus, we modified operator coefficients which are less sensitive to directions. Convolution operator is defined as 3×3 matrices, which is convolved with preprocessed image to produce a middle edge image. We define *MidEI* as a middle edge image applied by our edge operator. *MidEI* can be expressed by

$$MidEI(x,y) = C_1\sqrt{I_r(x,y)^2 + I_c(x,y)^2} \quad (1)$$

where C_1 represents a small constant. Constant C_1 in equation (1) causes vivid and thick edges that consist of boundaries of any object. Normally, I_r and I_c which can be interpreted as finding horizontal and vertical gradients at he point (x, y) respectively. In this work, we employed only the edge magnitude. To build edges thicker, our final edge image $ThickEI(x, y)$ is computed by

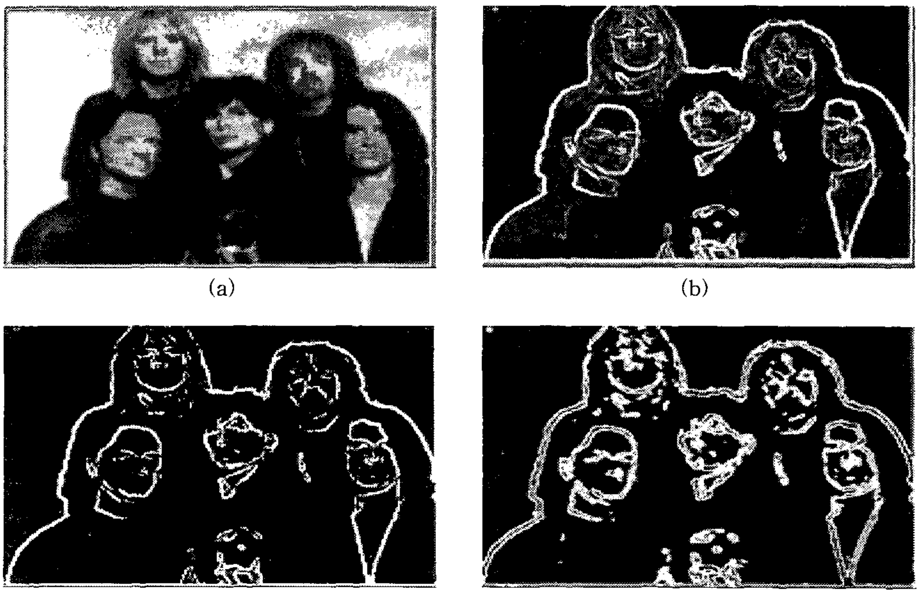
$$ThickEI(x,y) = C_2\sqrt{I_r(x,y)^2 + I_c(x,y)^2} \quad (2)$$

where $ThickEI(x, y)$ represents the thick edge image applied *MidEI* as a middle edge image to equation (2). The point (x, y) is on the *MidEI*. The value of C_1 and C_2 set below 1 in our system.

As shown in Fig. 2b, we applied an orignal image in Fig. 2a to the Sobel operator. We can see lots of noises caused by the sensitivity of directions. Noise is unwanted information that can result from the image acquisition process. These noises are difficult to discard windows containing non-faces before inputted to a classifier for detecting a face. Thus, we try to remove these noises using our edge filter. Fig. 2c and 2d illustrate *MidEI* image as a result of equation (1) and *ThickEI* image as a result of equation (2). Compared with the edge image for Sobel operator, *MidEI* image is removed lots of noises. As it can be observed, boundaries of each face in Fig. 2c have thicker than those of each face in Fig. 2b. In order not to miss the window containing a potential face in the input image, we utilize the devised edge filter that makes thick edges. We eliminate the areas that have the number of edges less than the given threshold value. The threshold value is chosen by experiments.

3.2 Face Detectors Using Neural Network

We have three face detectors using a typical three-layer back-error propagation neural network [7] and need to train each neural network. Back-error propagation has been the most widely used



(a) (b) (c) (d)

Fig. 2. (a) An original image ; (b) An edge image using Sobel operator ; (c) An *MidEI* image for the result of equation (1) ; (d) A *ThickEI* image for the result of equation (2).

of the neural networks paradigms and has been successfully utilized in applications such as image classification, sonar target recognition and pattern-analysis problems.

Back-error propagations of the system are layered with each layer fully connected to the above layer and the below layer. Initially, the weights are set to small-randomized values. The output of this detector is a real number.

One detector among three face detectors is to search for a whole face in the input image. We defined the detector as a whole face detector. We used a training set of 2000 face images that includes 1000 positive examples and 1000 negative examples, each 25×25 pixel window in size. This whole face detector using back propagation network is trained to classify 25×25 intensity images as face or non-face and produced real values between 1 to -1, indicating whether or not a window contains a face, respectively.

Component based face detector using a part of facial features is less sensitive to the images that have partial facial occlusion by interfering objects or strong directional lighting. We utilize part-based object detection schemes that have been proposed by many authors. All architectures of Component based systems need to select a part of face to use as an important feature. Some systems use features that seem naturally salient to humans such as the eyes, the nose, and the mouth. Other systems have been designed to automatically determine parts from the training images. The parts of the face are less sensitive than the whole face in various lighting conditions or face poses.

Therefore, we utilize the whole face detector for a whole face and component-based eyes detector for detecting each of the 12×12 eyes. We used the same whole face detector in phase one and phase two in our system. In the first phase, we employed *ThickEI* image as a input to the whole face detector on the purpose of removing the number of search windows containing non-faces.

When the search window is classified as a face, 25×25 intensity images in corresponding to the *ThickEI* image of the search window become an input to neural network filter in the second phase.

The eyes are trained to each left- and right-eye detector. We utilize each training set of each left- and right-eye 200 images including 100 positives and 100 negatives, each 12×12 pixel window in size. The hidden units for three detectors present a 1×5 pixel window. With the purpose of further accurate detection and verifying face candidates, we utilize three detectors that combined the whole face detector and the component-based eyes detectors. Each 12×12 pixel window containing potentially left-eye and right-eye candidate is passed through the left- and the right-eye detector respectively. After applying to two eye detectors,

$$CombinedNets = \alpha\times FNet + \beta\times LEyeNet + \gamma\times REyeNet \tag{3}$$

where *FNet* , *LEyeNet*, and *REyeNet* represent the detector for a whole face, the left- and the right- eye detector, respectively. α, β, and γ present a weighted constant for each detectors. In our system, α, β, and γ under experiment is 0.5, 0.25, and 0.25, respectively. We get *CombinedNets* that is calculated in (3). If the result in (3) is greater than a certain threshold, it can be detected a face. We show that the result of (3) can eliminate false positives. The threshold under experiment is over about 0.7.

4. EXPERIMENTAL RESULTS

Our system was tested with a large database of images gathered from CMU (Carnegie Mellon University) face test data sets composed of 130 still gray images with 507 faces. The images have various different sizes and are collected from photographs, newspaper pictures, the World Wide Web, and digitalized television pictures. Set A contains 42 images with 169 frontal view faces with 46927864 windows set B has 23 images with 155 faces with 8607480 and set C contains of 65 images with 183 faces including 112420576 windows. Many algorithms spent tremendous time on training their systems. Our system does not need to examine the number of whole windows for the test sets because the first step of the system eliminated plenty of windows containing non-face using our new edge filter. We know that the number of searched windows in our system is significantly smaller than a total number of windows given in Table 1. Those windows go through the whole face detector called *FNet*. This *FNet* helps improving the system performance by speeding up the running time.

Detection and error rate for our system are shown in Table 2. Results for *FNet* showed a larger number of false positives compared with *CombinedNets* and other systems in Table 3. It is because our systems do not consider the lighting correction in preprocessing step. The performance can be improved by applying illumination correction to the system.

We have compared performance in Table 3. The results using *CombinedNets* are higher than two methods [21] and *FNet*. However, the detection rate and the number of false positives of neural network-based detection by Rowely et al are better than our system. This is because their system was trained a huge training sets and passed through a post-processing step.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a frontal and near-frontal face detection algorithm. Our system consisted of two phases. As shown in chapter 4, Our system speeded up the running time by rejecting plenty of searching windows containing non-faces using our new edge filter. We developed a new edge filter which brings thick edges. It was very helpful in reducing the number of candidate windows including non-face.

In order to verify face candidates, we combined

Table 1. Number of 25×25 input windows to our system

	Test Set A	Test Set B	Test Set C	Total
No. of windows	46927864	8607480	112420576	167955920
No. of IW	85014	15593	203660	304267

IW: The number of input windows to the neural network filter after applying the our edge filter on the first step of our system.

Table 2. Detection and error rate for our system on the Test Set A, B, and C

Our system	Test Set A		Test Set B		Test Set C	
	Detect %	# False	Detect %	#False	Detect %	#False
FNet in phase one	86.4%	1816	74.6%	919	89.3%	1927
CombinedNets	90.2%	101	89.2%	83	91.4%	112

Table 3. Comparison results between face detection systems on Test Set B

System		Detect %	#False
Previous systems	Multi-layer network [16]	76.8%	5
	Perceptron [16]	81.9%	13
	Neural network-based detection [1]	92.9%	64
Our system	*FNet* in phase one	74.6%	919
	CombinedNets	89.2%	83

Fig. 3. Experimental Results using our system on still gray images from CMU Test data sets. The system misses some faces with significant illumination conditions and detects false positive examples. Note that the system didn't employ light correction.

three detectors called *CombinedNets*, which are the whole face detector for a whole face and component-based eye detectors for both eyes. Using *CombinedNets*, we obtain the high detection rates between 89.2% and 91.4 % of faces for three test sets provided by CMU. Fig. 3 shows example output images from *CombinedNets* on images from CMU Test data sets.

As future work, one of our goals is to detect a multi-view face detection using our system. The performance can be improved by applying proposed algorithm in order to reduce tremendous non-face candidates in various images.

6. REFERENCES

[1] H.A. Rowley, S.Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.20, No. 1, pp. 23-28, 1998.

[2] D. Roth, M.-H. Yang and N.Ahuja, "A SNow-based face detector," *Proc. Advances in Neural Information Processing Systems (NIPS'99)*, Denver, pp. 862-868. December 1999.

[3] E.Osuna, R.Freund and F. Girosi, "Training support vector machines: An application to face detection," *CVPR*, pp. 130-136, 1997.

[4] P. Viloa and M.J. Jones, "Robust real-time object detection," *IEEE ICCV Workshop on Statistical and Computational Theories of Vision*, Vancouver, Canada, 2001.

[5] K.K. Sung. "Learning and example selection for object and pattern detection," *PhD thesis, MIT lab*, 1996.

[6] Judith E., Dayhoff, *Neural network architectures: an introduction*, John Wiley and Sons Publisher, New York: Van Nostrand Reinhold, 1990.

[7] S. E UMBAUGH, *Computer vision and image processing*, Printice-Hall PTR., 1998.

[8] Antonio J. Colmenarez and Thomas S. Huang., "Face detection with information-based maximum discrimination," *Computer Vision and Pattern Recognition*, pp. 782-787, 1997.

[9] Bernd Heisele, Tomaso Poggio et al, "Face detection in still gray images," *MIT AI Memo-No.1687*, May, 2000.

[10] Rein-Lien Hsu; Abdel-Mottaleb, M.; Jain, A.K., "Face detection in color images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, Issue: 5, pp. 696-706, May 2002.

[11] Hjelmas, Erik and Low, Boon Kee, "Face Detection: A Survey.", *Computer Vision and Image Understanding*, Vol. 83, pp. 236-274, 2001.

[12] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade, "Rotation Invariant Neural Network-Based Face Detection," *Computer Vision and*

Pattern Recognition, pp. 38-44, 1998.

[13] H. Rowley, "Neural Network-Based Face Detection," *Ph.D. Thesis Carnegie Mellon University*, CMU-CS-99-117, 1999.

[14] S.H. Lin, S.Y. Kung, and L.J. Lin, "Face Recognition/Detection by Probabilistic Decision-Based Neural Network," *IEEE Trans. Neural Networks, Special Issue on Artificial Neural Networks and Pattern Recognition*, Vol. 8, No. 1, 1997.

[15] Ming-Hsuan Yang, David Kriegman, and Narendra Ahuja, "Detecting Faces in Images: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, Vol. 24, No. 1, pp.34-58, 2002.

[16] Kah-Kay Sung. "Learning and Example Selection for Object and Pattern Detection," *PhD thesis*, MIT AI Lab, Available as AI Technical Report 1572, 1996.

[17] Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," *CVPR*, 2001.

Young Sook Lee

She received her BS degree in computer engineering from the University of Dongseo, Busan, in 1996 and received the MS degree in computer science from Pukyong National University in 2000. She is currently a PhD student at Pukyong National University in computer science department. Her research interests include computer vision, machine learning, and computer animation.

Young Bong Kim

He received his BS degree in computer science and statistics from Seoul National University, Seoul, in 1987. He received the MS and PhD degrees in computer science from KAIST in 1989 and 1994, respectively. He is currently an associative professor in the department of electronics, computer and telecommunication Engineering at Pukyong National University. His current research interests are game programming, computer animation, and 3D object.