Application of Associative Memory in Human Face Detection

Mu-Chun Su and Chien-Hsing Chou
Department of Electrical Engineering, Tamkang University, Taiwan, R.O.C.
Email: muchun@ee.tku.edu.tw

Abstract

In this paper we present an associative-memory-based face detection system. First, the symmetry of human faces is used to quickly locate all the candidates of human faces with all possible sizes and locations. Then two associative memories are used to decide whether or not a human face exits at the locations. Some experimental results are given.

1. Introduction

In recent years the problem of human face recognition has attracted considerable attention [1]-[5]. A comprehensive overview of this problem can be found in [6]-[8]. The applications of face recognition systems widely range from secure access control, financial transactions to many others. The first important step of fully automatic human face recognition is human face detection. Face detection determines the locations and sizes of faces in an input image. Human faces represent one of the most common patterns in our vision. They are easily located in cluttered scenes by infants and adults alike. However automatic human face detection by computers is a very challenging task because face patterns can have significantly variable image appearances. For example, human faces vary from genders, ages, hair styles and races etc. In addition, the variations of scales, shapes and poses of faces in images also hinder the success of automatic face detection systems.

One of the earliest works in face detection was reported by Sakai et. al. [9]. They define the model of a human face in terms of several subtemplates corresponding to face contour, eye, nose, and mouth. An edge map extracted from the input image is matched against the subtemplates with possible variations in the position and size. The location of a face is determined from the scores of match. The merit of template matching techniques is that they are very simple. However, they prove to be inadequate because wide variations exist in human faces. Some face detection systems adopt image-invariance schemes that assume that there are certain spatial image relationships common and possibly unique to all face patterns [10]-[12]. Another approach to face detection is based on deformable templates that use parameterized curves and surfaces to model the nonrigid features of interests (e.g. eyes, mouth) of faces [13]. The template then interacts dynamically with the input image, by altering its parameter values to minimize deformation "stress" in the feature. Recently, the use of neural networks or other mechanisms in face detection has been studied by many researchers [14]-[17]. Training a neural network or a classifier for the face detection task is challenging because of the difficulty in characterizing prototypical "nonface images". Practically any image containing no face can serve as a nonface example. Since the space of nonface image is much larger than the space of face images how to collect a "representative" set of nonfaces so as to force the network or the classifier to learn the precise boundary between face and nonface image is a very demanding problem.

In this paper an efficient two-stage method to locate human faces in a complex background is proposed. We first use the symmetry property of human faces to quickly locate the candidate faces. Then two autoassociative memories are used to validate them. In Section 2, a brief review of associative memory is given. Section 3 gives the detailed discussion of our two-stage face detection algorithm. Some experimental results are given in Section 4. Conclusion is presented in Section 5.

2. Autoassociative Memory

Kohonen was the first to use an autoassociative memory to store and recall face image [18]-[19]. He demonstrated that an autoassociative memory could act as a content addressable memory for face image. In this paper we use two autoassociative memories to validate candidate faces. This section provides a brief description of the model of a linear autoassociative memory.

In a linear associative memory, the input-output relationship is described by

$$\underline{r} = W\underline{x} \qquad (1)$$

where the vector $\underline{x}$ and $\underline{r}$ denote the stimulus (input) and the response (output) of an associative memory. The matrix W is called the weight matrix that specifies the network connectivity in the associative memory. If the stimulus vector $\underline{x}$ equal to the response vector $\underline{r}$ the associative memory is referred to an autoassociative memory, otherwise, a heteroassociative memory. In this paper the inputs to the autoassociative memory are $N \times M$ gray level images. Any particular image (say the kth image) is represented by a column vector $\underline{x}_k$ of dimensionality $(N \times M) \times 1$. This vector consists of the concatenated rows of pixel intensities of the image.

Learning of the autoassociative memory may be achieved by using a simple Hebbian learning rule. This

0-7803-5529-6/99/$10.00 ©1999 IEEE

amounts to successively autoassociating each image vector, $\underline{x}_k$, and summing the resultant output-product matrices:

$$W = \sum_{k=1}^{K} \underline{x}_k \underline{x}_k^T \tag{2}$$

where W is an $(N \times M) \times (N \times M)$ matrix and the vectors are assumed normalized so that $\underline{x}_k^T \underline{x}_k = 1$. Let an input vector $\underline{x}_k$ be presented to the autoassociative memory, the response or retrieval of the memory is achieved by postmultiplying the matrix W by the input vector $\underline{x}_k$ as follows:

$$\begin{aligned} \underline{r} &= W \underline{x}_k \\ &= \underline{x}_k (\underline{x}_k^T \underline{x}_k) + \sum_{l \neq k} \underline{x}_l \underline{x}_l^T \underline{x}_k \\ &= \underline{x}_k + \sum_{l \neq k} \underline{x}_l \underline{x}_l^T \underline{x}_k \end{aligned} \tag{3}$$

When all the input vectors stored in the memory are mutually orthogonal, recall is perfect (i.e. $\underline{r} = \underline{x}_k$). If the input vectors are not orthogonal, the second term on the right-handed side of Eq.(3) is a "noise vector" that arises because of the crosstalk between the input vector $\underline{x}_k$ and all the other vectors stored in memory.

3. Two-Stage Human Face Detection Algorithm

Our human face detection algorithm involves in two stages. It first uses the symmetry property of human faces to quickly locate all candidate faces. In the second stage two autoassociative memories to validate candidate faces as faces or nonfaces.

3.1 Stage One: Locate Candidate Faces

Here we assume the human face can be approximated by an ellipse, therefore, our goal is to quickly locate ellipsoidal-shaped objects in an image. In fact it is not a trivial task since the sizes and orientations of the objects of interest may vary a lot. In order to efficiently solve the problem we adopt the following five-step procedure.

Step 1: Use the Sobel filter to find the edge image.
Step 2: Use a thinning algorithm [20] to thin the edge image.
Step 3: Delete short segments that contain less than five pixels.
Step 4: Use a $w_1 \times w_2$ window to scan the processed edge image from top to bottom and from left to right. We use the following symmetry function to measure the degree of symmetry of the object within the window relative to the center of the window. The symmetry function is defined as

$$d_s(\underline{x}_i, \underline{c}) = \min_{j \neq i} \frac{\| (\underline{x}_i - \underline{c}) + (\underline{x}_j - \underline{c}) \|}{\| (\underline{x}_i - \underline{c}) \| + \| (\underline{x}_j - \underline{c}) \|} \tag{4}$$

$$for\ 1 \le i \le N_w$$

where N_w is the number of edge pixels in the window, $\underline{c}$ represents the coordinates of the center of the window and $\underline{x}_i$ represents the coordinates of an edge pixel in the window. If $d_s(\underline{x}_i, \underline{c})$ is larger than a prespecified threshold θ then we increase N_c by one. Therefore, the final value of N_c represents the number of symmetrical pairs in the window. A detailed explanation about the symmetry function is given in [21].
Step 5: Sort the value of N_c in the decreasing order. Then the window with the largest N_c locates the most possible region containing a face.

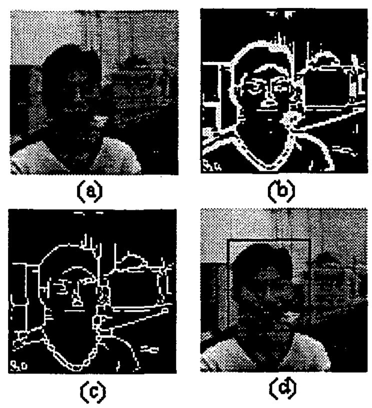


Fig.1. The processing steps in stage one: (a) original image, (b) the result after applying Sobel filter, (c) the thinned image, (d) the window with the largest N_c.

3.2 Stage Two: Using Autoassociative Memories to Validate Candidate Faces

We work with both the intensity image of the face as well as the edge image found using the Sobel filter. Stage two first requires the following training procedure:

Step 1: Acquire a set of face images (the training set).
Step 2: Calculate the weight matrix of the first autoassociative memory from the training set using Eq.(2).
Step 3: Apply the Sobel operator to the face images to acquire the edge images.
Step 4: Calculate the weight matrix of the second

autoassociative memory using the set of edge images.

Having trained two autoassociative memories the following step are then used to check if the face candidate is a face at all.

Step 1: Present the intensity image of the face candidate to the first autoassociative memory and compute the retrieval image using Eg.(3).

Step 2: Check if the retrieval image is sufficiently close to the face candidate. That is, if the cosine of the angle between the face candidate and the retrieval image is larger than a prespecified threshold θ_g then these two images are claimed to be close to each other.

Step 3: Apply the Sobel operator to the face candidate to acquire its edge image. Present the edge image to the second autoassociative memory and then compute the retrieval image using Eg.(3).

Step 4: Check if the retrieval image is again sufficiently close to the edge image. That is, if the cosine of the angle between the edge image and the retrieval image is larger than another prespecified threshold θ_e then these two images are claimed to be close to each other.

If the two computed distances are simultaneously below their respective thresholds then we declare the face candidate is really a human face, otherwise, a nonface. The reasons why we use two autoassociative memories are as follows. First there are many nonface images whose distributions of gray levels are similar to faces' distributions. Second, some objects have similar ellipsoidal-shaped contours as human faces. Fig. 2 gives examples to illustrate these observations. At last, it should be emphasized that we handle multiple scales by using windows of different sizes in stage one.

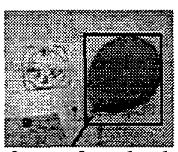


Fig. 2. An example of a nonface that has ellipsoidal-shaped contour.

4. Experimental Results

Our face detection algorithm was tested on two data sets. The training data set consists of a total of 20 image with a total of 20 face patterns collected by our lab. The testing data set consists of a total of 30 images with 35 face patterns partly collected by our lab and partly collected at CMU [22]. We choose 20 face images from the training data set to train two autoassociative memories. Fig. 3 gives some examples of the face image used to train the associative memories. These face images were rescaled to 41×50 image. The two thresholds θ_g and θ_e were chosen to be 0.28 and 0.1, respectively. We then used the 30 images with 35 face patterns from the testing data set to test our algorithm. 32 faces were identified. Fig. 4(a), Fig 5(a), Fig. 6(a), Fig. 7(a), Fig. 8(a) and Fig. 9(a) show the results of stage one in our algorithm. Fig. 4(b), Fig 5(b), Fig. 6(b), Fig. 7(b), Fig. 8(b) and Fig. 9(b) show the results of stage two in our algorithm.

Fig. 3. Example of face images used in the training set.

(a) (b)

Fig. 4. A baby face in the image.

(a) (b)

Fig. 5. A human face in the image .

(a) (b)

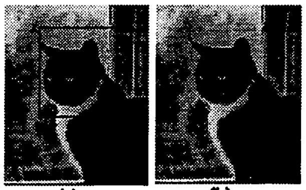
Fig. 6. A face of a puppet in the image.

(a) (b)

Fig. 7. Two human faces in the image.

(a) (b)

Fig. 8. Two human faces and a ballon in the image.



(a) (b)

Fig. 9. A cat in the image.

5. Conclusion

We have developed an algorithm to detect human faces. The algorithm consists of two steps: using the symmetry property of human face to quickly locate candidate region containing human faces and using two autoassociative memories to validate them. The most appealing advantage of our algorithm is that the training time of the two autoassociative memories is very short compared to other neural network-based algorithms. The experimental results were encouraging.

References

[1] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," IEEE Trans. Patt. Anal. And Mach. Intell., vol. 15, pp. 1042-1052, 1993.

[2] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human face," IEEE trans. Patt. Anal. And Mach. Intell., vol. 12, pp. 103-108, 1990.

[3] M. Turk and A. Pentland, "Eigenfaces for recognition," Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991.

[4] L. Harmon and W. hunt, "Automatic recognition of human face profiles," Computer Graphic and Image Process., vol. 6, pp. 135-156, 1997.

[5] T. Kanade, Computer Recognition of Human Faces, Basel and Stuttgart: Birkhauser, 1997.

[6] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: a survey," Proceedings of the IEEE, vol. 83, no. 5, pp. 705-740,1995.

[7] D. Valentin, H. Abdi, A. J. O'toole and G. W. Cottrell, "Connectionist models of face processing a survey," Patt. Recog., vol. 27, no. 9, pp. 1209-1230, 1994.

[8] A. Samal and P. Iyengar, "Automatic recognition and analysis of human faces and facial expressions: A survey," Patt. Recog., vol. 25, pp. 65-77, 1992.

[9] T. Sakai, M. Nagao, and S. Fujibayashi, "Line extraction and pattern recognition in a photograph," Patt. Recog., vol. 1, pp. 233-248, 1969. [10] G. Yang and T. S. Huang, "Human face detection in a complex background," Patt. Recog., vol. 21, no. 1, pp. 53-63, 1994.

[11] V. Govindaraju and M. Rajapakse, "Verification of faces in cluttered photographs," IEEE International Conference on Systems, Man, and Cybernetics, pp. 2028-2033, 1997.

[12] P. Sinha, "Object recognition via image invariance a case study," Investigative Ophthalmology and Visual Science, vol. 35, pp. 1735-1740, 1994.

[13] A. Yuille, P. Hallinaw, and D. Cohew, "Feature extraction from faces using deformable templates, " Int'l J. Computer Vision, vol. 8, no. 2, pp. 99-111, 1992.

[14] P. Seitz and G. K. Lang, "Using local orientation and hierarchical spatial feature matching for robust recognition of objects," Proceedings of SPIE, pp. 252-259, 1991.

[15] S.-H. Lin, S.-Y. Kund, and L.-J. Lin, "Face recognition / detection by probabilistic decision-based neural network," IEEE Trans on Neural Networks, vol. 8, no. 1, pp. 114-132, 1997.

[16] H. A. Rorvley, S. Baluja, and T. Kanade, "Neural network- based face detection," IEEE Trans. on Patt. Anal. And Mach. Intell., vol. 20, no. 1, pp. 23-38, 1998.

[17] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," IEEE Trans on Patt. Anal. And Mach. Intell., vol. 20, no. 1, pp. 39-51, 1998.

[18] T. K. Kohonen, Associative Memory: A System Theoretic Approaches, Springer, Berlin, 1977.

[19] T. Kohonen, E. Oju, and P. Lehtio, "Storage and processing of information in distributed associative memory systems," Parallel Models of Associative Memory, G. E. Hinton and J. A. Anderson, editors, pp. 49-81. Erlbaum, Hillsdale, 1981.

[20] R. C. Gonzalez and R. E. Woods, Digital Image Processing, Addison-Wesley Publishing Company, 1989.

[21] M.-C. Su and C.-H. Chou, "A Competitive Learning Algorithm Using Symmetry," to be published in IEICE.

[22] These image are available over the World Wide Web, at the URL http://www.ius.cs.cmu.edu/IUS/usrp0 /har/FaceDemo/ gallery-inline.html.