Expand Training Set for Face Detection by GA Re-sampling

Jie Chen[1] Xilin Chen[1] Wen Gao[1,2]
[1]School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150001, China
[2]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, China
chenjie@jdl.ac.cn, xlchen@ieee.org, wgao@jdl.ac.cn

Abstract

Data collection for both training and testing a classifier is a tedious but essential step towards face detection and recognition. All of the statistical methods suffer from this problem. This paper presents a genetic algorithm (GA)-based method to swell face database through re-sampling from existing faces. The basic idea is that a face is composed of a limited components set, and the GA can simulate the procedure of heredity. This simulation can also cover the variations of faces in different lighting conditions, poses, accessories, and quality conditions. All the collected face samples are aligned and randomly divided into three sub-sets: training, validating, and testing set. The training set is then used to train a Sparse Network of Winnow (SNoW). In addition, it is also used as the initial population of the GA. After each generation, we will use the initial generation and the solutions with high fitness values to re-train the SNoW, and the newly-trained SNoW is used to evaluate the individuals of next generation and also tested on validation set and test set. To verify the generalization capability of the proposed method, we also use the expanded database to train an AdaBoost-based face detector and test it on the MIT+CMU frontal face test set. The experimental results show that the data collection can be speeded up efficiently by the proposed methods.

1. Introduction

Over the past ten years, face detection has been thoroughly studied in computer vision research for its interesting applications. Face detection is to determine whether there are any faces within a given image, and return the location and extent of each face in the image if one or more faces present [15]. Recently, the emphasis has been laid on data-driven learning-based techniques. Sung and Poggio presented an example based learning method with six Gaussian clusters to model the distributions for face and nonface patterns respectively [12]. The density functions of the distributions were then fed to a multiple layer perceptron for face detection. Rowley et al. developed a neural network-based face detection system to examine small windows of an image and decide whether each window contains a face [9]. In order to detect faces with rotation in the image plane, the system was extended to incorporate a separate router network to determine the orientation of the face pattern [10]. Schneiderman and Kanade proposed a face detector based on the estimation of the posterior probability function, and profile images were added to the training set to incorporate such statistics to detect side views of a face [11]. Yang et al. proposed a method that used a SNoW learning architecture to detect faces with different features and expressions, in different poses, and under different lighting conditions [14]. Liu presented a Bayesian Discriminating Features (BDF) method for multiple frontal face detection, which was trained on images from only one database, yet worked on test images from diverse sources, displayed robust generalization performance [7]. Viola described a rapid object detection scheme based on a boosted cascade of simple features. It brought together new algorithms, representations and insights which could broader applications in computer vision and image processing [13]. Li et al. proposed a FloatBoost-based algorithm to guarantee monotonicity of the sequential AdaBoost learning and developed a real-time multi-view face detection system [6].

The performance of these learning-based methods highly depends on the training set [8], and they suffer from a common problem of data collection for training. This paper focuses on this problem. We propose a re-sampling method to generate more samples from existing ones by using GA operations. The intuitive idea is that a face is composed of limited types of components, and we use the GA to simulate the procedure of heredity.

The rest of this paper is organized as following: Section 2 presents the details of the GA-based database expanding. The experiment results of GA are described in Section 3. In Section 4, we give the conclusions.

2. Re-sampling using Genetic Algorithms

The system overview is given in Fig. 1. Initially, all collected images are aligned coarsely, preprocessed and divided into three sets: the training, validating and testing set. The training set is employed as the initial population to perform the GA operations. All the

Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR'04)
0-7695-2122-3/04 $ 20.00 © 2004 IEEE

IEEE COMPUTER SOCIETY

intermediate solutions are evaluated by a classifier SNoW trained by the initial population and the non-face samples. Fitter solutions survive while weaker ones perish. All the survival solutions and the initial population are composed of the next population which is utilized to re-train the classifier SNoW again. The newly-trained classifier is used to evaluate the next intermediate solutions. If the termination criterion is reached, the iterative GA operations are stopped and the last population is the ultimate solutions.

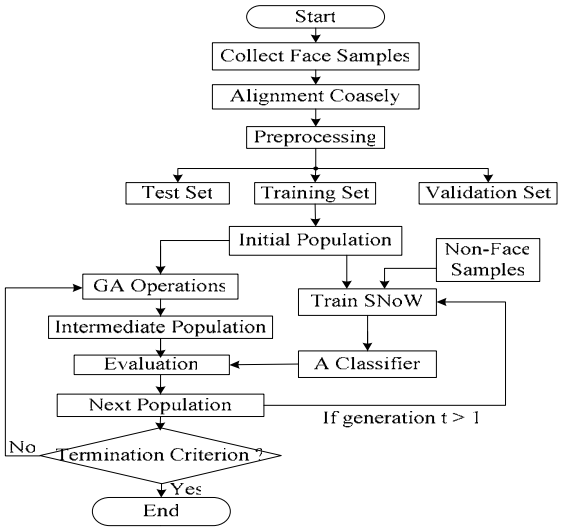

Fig. 1. Overview of the system.

2.1. Face-Samples Preprocessing

First, we align all the collected face samples to reduce the extrinsic variations among them. In our method, we apply the preprocessing proposed by Rowley et al [9] to align face samples. To make the detection method less sensitive to affine transform, the images are often rotated, translated and scaled [1, 3, 6, 7, 9 10, 14]. Before GA operations, therefore, we randomly rotate these samples up to $\pm 15^{\circ}$, translate up to half a pixel, and scale up to ±10%. Then, histogram equalization is performed, which maps the intensity values to expand the range of intensities. The same procedure (histogram equalization) is applied to the initial population and all successive reproduced populations. By these preprocessing, the solutions fitness can be improved.

The face-image database consists of 6,000 faces (collected form Web) which cover wide variations in poses, facial expressions and lighting conditions. After these preprocessing, we get 30,000 face images, and then this database is randomly divided into three sets: training set (which consists of 15,000 images), validation set (5,000 images) and test set (10,000 images).

2.2. Genetic Algorithms

Genetic algorithms take their analogy from nature [4]. They create an initial population, often represented as bit strings, which evolve over successive generations. Those individuals which represent a better solution to the target problem are given more chances to "reproduce" than those individuals which are poorer solutions. They are mated with other solutions by crossing parts of a solution string with another. The reproduced strings are also mutated. The schematic of GA operations are shown in Fig. 2. Over time, they reproduce by crossing high fitness solutions at random points to weed out poor fitness solutions. These operations act to randomly-sample a large part of the huge state space very efficiently. Holland first proposed the artificial reproduction schemes in 1970s [4].

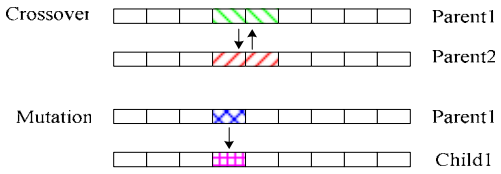

Fig. 2. The schema of crossover and mutation.

The main procedure of the genetic algorithm is:

. Encoding. As discussed in [14], assume that the pixel at (x, y) of an image, with width w and height h, has intensity value $I(x, y)$ $0 \le I(x, y) \le 255$. This information is encoded as a string whose index is: $l(i) = l(y \times w + x) = 256(y \times w + x) + I(x, y)$. In our experiments, the values for w and h are 20, since each face sample has been normalized to an image of 20×20 pixels. A gene in an individual can be denoted by $l(i)$ $(0 \le i \le 400)$, and an individual A is represented as a string $(l_1)(l_2)(l_3)\cdots(l_i)\cdots(l_{400})(w_j)$, where w_j is the fitness value of this individual.

. Initial Population. In general, the initial population is randomly generated [4]. In this method, however, we will end up with a population where each individual is generated by encoding a normalized face sample of 20×20 pixels, and it is actually the training set as discussed in Section 2.1.

. Crossover and Mutation. In our scheme, we consider "1-point" crossover in order to manipulate the fitness of solutions. Furthermore, every two parents crossover at fixed locus. That is to say, we will break down each parent into smaller pieces without overlapping: forehead, eye, nose, mouth, as demonstrated in Fig. 3 (a), and the process of crossover are shown in Fig. 3 (b). Mutation, in our method, is accomplished by sharpening or blurring or lighting.

IEEE COMPUTER SOCIETY

The procedure of sharpening or blurring is: First of all, a sub-image, about a quarter to half size of its parent, is obtained from its parent, then it is sharpened or blurred randomly, and then we recombine the changed sub-image and the unchanged part to reproduce its child. To avoid the trace resulting from recombination, the intermediate solution is smoothed as shown in Fig. 3 (c). As to the lighting, we use the same scheme as that in [5].

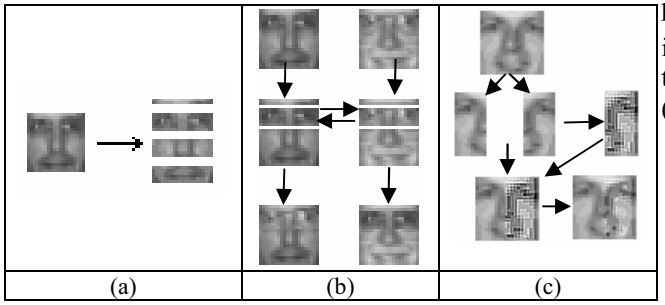

Fig.3. Crossover and mutation. (a) Each parent is converted into a sequence of observation vectors for crossover, (b) Crossover, (c) Mutation

. Fitness Evaluation. The fitness function, used to evaluate the fitness of a solution, is a classifier called Sparse Network of Winnows (SNoW) [14]. To train this classifier, we use a new feature pattern [1] to characterize the samples as demonstrated in Fig. 4. It encodes a local block of each 20×20 sample on spatially pixel intensities. The pixel intensities and their coordinates at each possible 2×2-block are quantized into one scalar value. That is to say that the 4 grey pixels and its position are all coarsely encoded using 3-bit resolution, and we can get an 18-bit resulting string as shown in Fig. 4. Before the calculation of the feature, histogram equalization is applied to the sample. And then we get 2^{18}=262,144 possible binary features, which enlarge the feature space further and make the two class problem (face and non-face) more linear separable when compared with the feature space in [14] (102,400 possible binary features). Each 20×20-pixel sample has exactly 19×19 =361 feature components and each is connected to a 2×2-block. Each of these 361 feature components addresses a single weight vector of the classifier. The sum over all weight components addressed by the current feature vector matches a normalized score. (Note this score is assigned to the input face sample as its fitness.) The classifier contains 262,144 weights in total. They are trained using the SNoW training procedure. For the details of the SNoW-classifier, please refer to [14].

To train the evaluation function SNoW (or classifier) of each generation, we use the initial population and the solutions of the last generation as positive samples as illustrated in Fig. 1. For negative examples we start with 15,394 non-face examples from 6,107 images of landscapes, trees, buildings, etc. Although it is extremely difficult to collect a typical set of non-face examples, the bootstrap [12] is used to include more non-face examples during training. And each resulting classifier is used to test on the validation set and evaluate the successive generation. After each solution is evaluated, a fitness value is attached. As to the initial population, let the fitness value w_j =1 for each individual. Note the second generation is evaluated by the classifier trained only by the initial population (about 15,000 face images) and the negative examples.

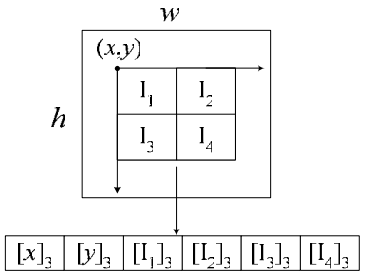

Fig. 4. Quantization method of grey features.
Where (x, y) is the coordinate of the 2×2-block; I_j (j=1, 2, 3, 4) are 4 pixels in this block; the subscript "3" in $[x]_3$, $[y]_3$ and $[I_j]_3$ denotes each component is encoded by 3-bit resolution in the feature representation.

2.3. Re-sampling

The initial population, (which actually is the training set in our method and it contains 15,000 face images), is divided into several smaller sub-sets according to the rotating angle of each image. Here, we divide all these images evenly into six sub-sets: the first is those within $[-15^\circ,-10^\circ)$ and denoted by ω_1 as shown in Fig. 5; the 2nd is those within $[-10^\circ,-5^\circ)$ and denoted by ω_2;…; and the last is those within $[10^\circ,15^\circ]$ and denoted by ω_6. Then all these six sub-sets are used as the initial population of the GA operations.

Here, we use the "roulette selection" to choose individuals. The roulette selection is based on the fitness value of each individual — the higher an individual's fitness is, the more chances it has. As shown in Fig. 5, these selected individuals are put into mating pools, and those individuals within the same sub-set will crossover with the probability P_c. For example, two individuals, x_i and x_j, selected from ω_6, are put into Pool6. After the crossover operations, their offspring will be put into ω_6 again. It is the same that those individuals selected from ω_5 are put into Pool5

IEEE COMPUTER SOCIETY

for crossover, and their offspring are returned to ω_5, and so on as demonstrated in Fig. 5. Some parents of the current generation (not their children in our method) mutate with the probability P_m. For example, x_k selected from ω_1 is mutated and its child is laid back into ω_1.

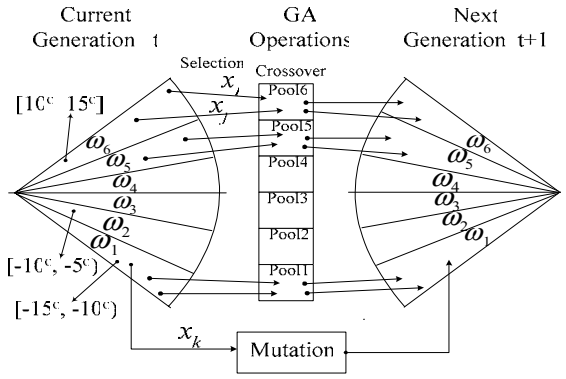

Fig. 5. The process GA operations.

After each generation reproduced, we keep 10% children with high fitness and discard the others. In this scheme, after every 20 generations, the population will contain $15,000\times(1+0.1)^{20}=100,912$ individuals. Its size is much larger than that of the original one. In order to keep it in control, we cut down the scale and keep it within 3 times of the initial generation. That is to say we keep only 45,000 individuals, which including the 15,000 initial population and their 30,000 children. All the remained children of every 20-generation are checked manually to avoid the classifier assigning a biased fitness value, which means some solutions may have high fitness value assigned by the classifier but do not look like faces.

After every one generation, we use its solutions (together with the initial population) and the negative set as new training set to train the classifier SNoW as demonstrated in Fig. 1. Note the newly-trained classifier is used to evaluate the solutions of next generation. And the newly-trained SNoW is also tested on the validation set (as discussed in Section 2.2). Moreover, we compare these results of each generation. The GA operations will be terminated when their resulting difference between two neighbor generations drops below a predetermined threshold. Some solutions by the GA operations are shown in Fig. 6.

Fig. 6. Some face samples generated by the GA re-sampling.

3. Experiment

3.1. Comparing the solutions performance of the different generations

Fig. 7 provides the results for the classifier SNoW trained with different database (based on the different generations) and tested on validation set. In this figure, we use only the initial population (NoGA), and the initial population together with the solutions of the 20th generation (GA20) or the 40th, or the 60th, or the 80th generation (the same as GA40, GA60, GA80) as face-sample sets. It means the NoGA has 15,000 face samples, while GA20, GA40, GA60 or GA80 has 45,000 face samples (including the initial population and their solutions respectively). For all of these five cases, the trained classifiers are tested on the validation set.

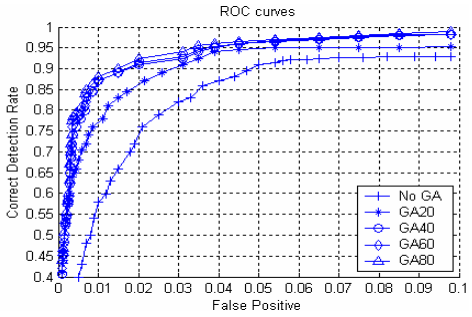

Fig. 7. The ROC curves on the validation set using different generations of GA as training set for a fixed classifier.

Form these Receiver Operating Characteristic (ROC) curves in Fig. 7, we can find that the performance of GA20 is much better than that of NoGA, and GA40 improve the result further. However, the difference between GA60/GA80 and GA40 is very limited. Furthermore, we check these five trained classifiers by the test set to verify their generalization performance. We find that they will obtain almost the same results compared to test on the validation set. That is to say that GA20 also outperforms NoGA, and GA40 works better than GA20, while GA40, GA60, GA80 almost have similar results on the test set. From these results, we can conclude that the proposed algorithm has good generalization performance.

In addition to comparing the solutions performance of different generations, we also investigate some possible reasons for the success of GA in this domain. First, the selection scheme gives more chances to those typical samples; and using these samples, we can generate more samples with representative features by GA. Second, we can acquire some variations about one person by crossover. For example, we can make one with glasses or beard to simulate the variations of a person, and these variations

IEEE COMPUTER SOCIETY

are very common in our daily life. Third, the mutation can simulate the variations from the lighting and quality conditions of an image to the aging and makeup of a person. Therefore, it can gain the diversities of the samples.

3.2. Evaluation of the generated samples

3.2.1. The AdaBoost-based classifier. Considering that all the solutions of each generation are evaluated by the classifier SNoW during expanding, they may favor this classifier. In order to verify that the solutions are independent to any special classifier, we use the expanded training set to train another classifier and test its generalization performance.

- Given example set ***S*** and their initial weights ω_1;
- Do for t=1,……,T:
  1. Normalize the weights ω_t;
  2. For each feature, j, train a classifier h_j with respect to the weighted samples;
  3. Calculate error ε_t, choose the classifier h_t with the lowest error and compute the value α_t;
  4. Update weights ω_{t+1};
- Get the final strong classifier: $h(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$.

Fig. 8. The AdaBoost algorithm for classifier learning.

A large number of experimental studies have shown that classifier combination can significantly exploit the discrimination ability comparing with those individual feature and classifiers. Boosting is one of the common used methods for combining classifiers [2]. AdaBoost, a version of the boosting algorithm, has been used in face detection and is capable of processing images extremely rapidly while achieving high detection rates [13]. Therefore, we use the AdaBoost algorithm to train a classifier. A final strong classifier is formed by combining a number of weak classifiers which is described in Fig. 8 following the notation in [2]. For the details of the AdaBoost based classifier, please refer to [2].

3.2.2. Training the detector. To compare the performance improvement on different training sets, we use three different face training sets. The first group consists of the initial population as discussed in Section 2.2. The second group contains a set of 45,000 face images generated by GA40 ***automatically*** which includes the initial population (15,000 faces) and its solutions (30,000 faces). The third group also contains a set of 45,000 face images generated by GA40 ***manually*** which includes the initial population (15,000 faces) and its solutions (30,000 faces). Here, the word "***automatically***" means that each solution of every one generation is evaluated by the classifier SNoW, while all the remained children of every 20-generation are checked manually (see Section 2.3 for details). The word "***manually***" means that both each solution of every one generation and all the remained children of every 20-generation are checked manually. Applying these two-group face sets: the group by GA40 ***automatically*** and the group by GA40 ***manually***, the aim is to compare the performance difference between by automatic manners and by hands.

The non-face class is initially represented by 5,000 non-face images. Each single classifier is then trained using a bootstrapping approach similar to that described in [12] to increase the number of images in the non-face set. The bootstrapping is carried out several times on a set of 8,736 images containing no faces. Note that non-face samples used to train the AdaBoost-based classifier are not the same with the non-face samples used to train the classifier SNoW.

3.2.3. Detection Results. The resulting detectors, trained on three different sets, are evaluated on the MIT+CMU frontal face test set which consists of 130 images showing 507 upright faces [9]. Some results are shown in Fig. 9. The detection performances on this set are compared in Fig. 10. From the ROC curves one can find that we get the detection rate of 90.48% and 12 false alarms with the detector trained on the set by GA40 ***automatically***. P. Viola reported a similar detection capability of 89.7% with 31 false detects (by voting) [13]. However, different criteria (e.g. training time, number of training examples involved, cropping training set with different subjective criteria, execution time, and the number of scanned windows in detection) can be used to favor one over another which will make it difficult to evaluate the performance of different methods even though they use the same benchmark data sets [15]. While using the detector, trained on the set by GA40 ***automatically***, in real-time applications, it can run at 16 frames per second averagely on a PC (with a Pentium III CPU @ 866MHz) when the image size is 320×240.

From the ROC curves we can also conclude that the detector trained on the set by GA40 ***automatically*** outperforms the detector trained on the set by GA40 ***manually***, expect two points in the curves (One point is when the false detects equal to 103, and the other is 226.). The possible reasons are: The first is that those face samples, selected according to one's subjective criteria, are not always optimal to train a detector. The second is that the subjective criteria are not always the same during the tedious work when one selects a sub-set from thousands of candidates. For example, in our experiments, we choose 50% individuals of the initial generation to crossover and 10% to mutate (The

IEEE COMPUTER SOCIETY

fraction for mutation is much smaller than that of the crossover, which is to avoid the quality deteriorism of the newly generated faces.). We let P_c=0.6 and P_m=0.1. After every one generation, we are needed to pick out 1,500 samples from 4650 candidates, and this operation is repeated by 40 times.

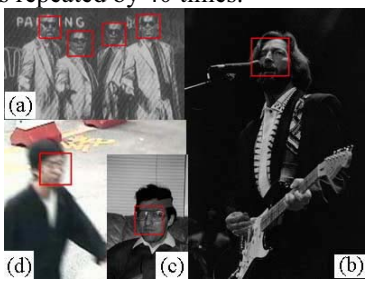

Fig.9. Some face detection results; (a), (b), (c) from MIT+CMU frontal face test set and (d) from the practical application of this system.

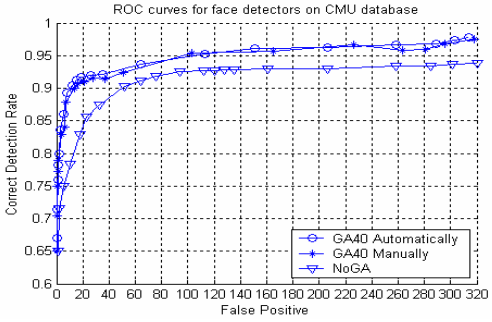

Fig.10. The ROC curves for our detectors on the MIT+CMU frontal face test set.

4. Conclusions

In this paper, we present a novel method to expand face sample set by applying the genetic algorithms. It can generate new face samples by crossover and mutation operations. These new generated samples can cover a widely variations: simulate the variations of faces in daily life and the variations of the images of lighting and quality conditions. We use some face samples without GA and those with GA but different generations to train a classifier SNoW, and compare the results tested on a validation set and an independent test set with these trained classifiers. The performances of the detector trained by both the initial generation and the solutions of GA are much better than that trained only by the initial generation. When the generation is up to 40, the result will approach a relatively stable result. Finally, we use the expanded face set by GA40 to train an AdaBoost-based classifier and test it on the MIT+CMU frontal face test set, and a detection rate of 90.48% is achieved while only 12 false alarms. It demonstrates that the expanded face samples set can be used to train other classifiers other than SNoW and can improve further the classifier performance.

5. Acknowledge

This research is partially sponsored by Natural Science Foundation of China under contract No.60332010, National Hi-Tech Program of China (No. 2001AA114190 and 2002AA118010), and ISVISION Technologies Co., Ltd.

References

[1] B. Fröba and A. Ernst. Fast Frontal-View Face Detection Using a Multi-Path Decision Tree. *In Proc. Audio- and Video-based Biometric Person Authentication (AVBPA '2003)*, 2003. pp. 921-928.

[2] Y. Freund and R. E. Schapire. *A decision-theoretic generalization of online learning and an application to boosting*. In Computational Learning Theory. 1995. pp. 23–37. Springer-Verlag.

[3] B. Heisele, T. Poggio, and M. Pontil. Face Detection in Still Gray Images. *CBCL Paper #187*. Massachusetts Institute of Technology, Cambridge, MA, 2000.

[4] J. H. Holland. Adaptation in natural and artificial systems. *Univ. of Michigan Press.* Reprinted in 1992, MIT Press.

[5] V. Krueger. Gabor Wavelet Networks for Object Representation. Ph.D. Thesis, Christian-Albrecht University, Kiel, Germany, Jan. 2001.

[6] S. Z. Li, L. Zhu, Z.Q. Zhang, A. Blake, H. J. Zhang, and H. Shum. Statistical Learning of Multi-View Face Detection. *In Proceedings of the 7th European Conference on Computer Vision*. 2002.

[7] C. J. Liu. A Bayesian Discriminating Features Method for Face Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, June 2003. pp. 725-740.

[8] X. Lu and A. K. Jain. Resampling for Face Recognition, *Proc. of 4th Int'l Conf. on Audio- and Video-Based Biometric Person Authentication*, 2003. pp. 869-877.

[9] H. A. Rowley, S. Baluja, and T. Kanade. Neural Network-Based Face Detection. *IEEE Tr. Pattern Analysis and Machine Intel.* vol. 20, 1998. pp. 23-38.

[10] H. A. Rowley, S. Baluja, and T. Kanade. Rotation Invariant Neural Network-Based Face Detection. *Conf. Computer Vision and Pattern Rec.*, 1998. pp. 38-44.

[11] H. Schneiderman and T. Kanade. A Statistical Method for 3D Object Detection Applied to Faces. *Computer Vision and Pattern Recognition*, 2000. pp. 746-751.

[12] K. K. Sung, and T. Poggio. Example-Based Learning for View-Based Human Face Detection. *IEEE Trans. on PAMI* Vol.20. , No. 1, 1998. pp. 39-51.

[13] P. Viola and M. Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. *Conf. Computer Vision and Pattern Recognition*, 2001. pp. 511-518.

[14] M. H. Yang, D. Roth, and N. Ahuja. A SNoW-Based Face Detector. *Advances in Neural Information Processing Systems 12*, MIT Press, 2000. pp. 855-861.

[15] M. H. Yang, D. Kriegman, and N. Ahuja. Detecting Faces in Images: A Survey. *IEEE Tr. Pattern Analysis and Machine Intelligence*, vol. 24, Jan. 2002. pp. 34-58.

IEEE COMPUTER SOCIETY