Resampling for Face Detection by Self-Adaptive Genetic Algorithm

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

Over the past ten years, face detection has been thoroughly studied in computer vision research for its interesting applications. However, all of the state-of-the-art statistical methods suffer from the data collection for training a classifier. This paper presents a self-adaptive genetic algorithm (GA)-based method to swell face database through re-sampling from the 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. 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 efficiently speeded up by the proposed methods.

1. Introduction

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 [18]. 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 [15]. 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 [12]. In order to detect faces with rotation in the image plane, the system was extended to incorporate a separate router network [13]. Schneiderman and Kanade proposed a face detector based on the estimation of the posterior probability function [14]. Yang et al. proposed a method that used a Sparse Network of Winnow (SNoW) learning architecture to detect faces with different features and expressions, in different poses, and under different lighting conditions [17]. Liu presented a Bayesian Discriminating Features (BDF) method for multiple frontal face detection which had robust generalization performance [9]. Viola described a rapid object detection scheme based on a boosted cascade of simple features [16]. Li et al. proposed a FloatBoost-based algorithm to guarantee monotonicity of the sequential AdaBoost learning [8].

The performance of these learning-based methods highly depends on the training set [10], 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 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 self-adaptive GA

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 intermediate solutions of the current generation are evaluated by a classifier SNoW trained by the last generation 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.

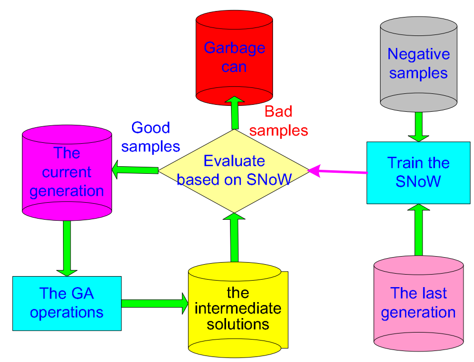


Fig. 1. Overview of the system.

2.1. Face-Samples Preprocessing

First, we align all the collected face samples to reduce the extrinsic variations among them. To make the detection method less sensitive to affine transform, the images are often rotated, translated and scaled [4]. Before the GA operations, 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 is also applied to the initial population and all successive reproduced populations.

The face 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. Self-adaptive genetic algorithms

Genetic algorithms take their analogy from nature and Holland first proposed the artificial reproduction schemes in 1970s [5]. The main procedure of the genetic algorithm is:

□. Encoding. As discussed in [17], Let the pixel at (x, y) of an image, with width w and height h, has intensity

Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)
1051-4651/04 $ 20.00 IEEE

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 $(l_i)\quad (0 \le i < 400)$, and an individual $\boldsymbol{A}$ 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 this method, the initial population is ended up with a population where each individual is generated by encoding a normalized face sample in the training set.

□. 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 with a probability $\boldsymbol{P}_c$ That is to say, we will break down each parent into smaller pieces without overlapping: forehead, eye, nose, mouth, as demonstrated in Fig. 2 (a), and the process of crossover are shown in Fig. 2 (b).

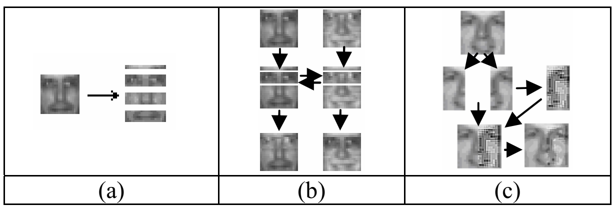


Fig.2. Crossover and mutation. (a) Each parent is converted into a sequence of observation vectors, (b) Crossover, (c) Mutation

Mutation, in our method, is accomplished by sharpening, blurring or lighting with the probability $\boldsymbol{P}_m$. 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 yielded by recombination, the intermediate solution is smoothed as shown in Fig. 2 (c).

As to the lighting of mutation operator, we use two kinds of strategies. One is the same as mentioned in [6] which is applied to simulate linear point light source; the other is the same as mentioned in [7] which is used to simulate more complex diffuse light fields by a configuration of nine point light source directions.

As discussed in [7], assuming a face is a convex Lambertian surface, we get the face image:

$$I(x,y)=\rho(x,y)\vec{n}(x,y)\vec{s}\,, \tag{1}$$

where $\rho(x,y)$ is the albedo of the point (x,y), $\vec{n}(x,y)$ is the surface normal direction and $\vec{s}$ is the point light source direction whose magnitude is the light source intensity.

In space-frequency domain, Lambertian surface is a low-pass filter and the set of images of a Lambertian object under varying lighting can be approximated by a 9D linear subspace spanned by harmonic images b_{lm} ($0 \le l \le 2$)[1]. The harmonic images are defines as:

$$b_{lm}(x,y)=\rho(x,y)A_l Y_{lm}(\theta(x,y),\phi(x,y)), \tag{2}$$

where Y_{lm} is spherical harmonic at the surface normal, (θ,ϕ) corresponding to pixels is (x,y) and A_l is the spherical harmonics coefficients. The image under arbitrary lighting can be written as:

$$I(x,y)=\sum_{l=0}^{2}\sum_{m=-l}^{l} L_{lm} b_{lm}, \tag{3}$$

where L_{lm} is the spherical harmonic coefficients of the specific lighting. The lower nine spherical harmonic coefficients L_{lm} ($0 \le l \le 2$) can be estimated as discussed in [11]. Once we have estimated the lighting of the original image, it is straightforward to relight it to the canonical illumination with illumination ratio image. Illumination ratio image between the canonical image and original image the is defined as:

$$IRI(x,y)=\frac{I_{can}(x,y)}{I_{org}(x,y)}=\frac{E_{can}(\theta(x,y),\phi(x,y))}{E_{org}(\theta(x,y),\phi(x,y))}\,, \tag{4}$$

where the subscripts are index of illumination, and E is the incident irradiance. Image relighting with illumination ratio image can be written as:

$$I_{can}(x,y)=IRI(x,y)\times I_{org}(x,y)\,. \tag{5}$$

In order to search the solutions effectively, the probability $\boldsymbol{P}_c$ and $\boldsymbol{P}_m$ are modulated self-adaptively. The modulation scheme is:

$$P_c=\begin{cases} k_1(f_{\max}-f_c^{'})/(f_{\max}-\bar{f})\,, & f_c^{'} \ge \bar{f} \\ k_3\,, & f_c^{'} < \bar{f} \end{cases} \tag{6}$$

$$P_m=\begin{cases} k_2(f_{\max}-f)/(f_{\max}-\bar{f})\,, & f \ge \bar{f} \\ k_4\,, & f < \bar{f} \end{cases} \tag{7}$$

where $f_c^{'}$ is the bigger one of two parents' fitness value; f is the fitness of the mutated parent; $f_{\max}$ is the maximum fitness value of the current population and $\bar{f}$ is its average fitness; $K_1=k_3=1$, $k_2=k_4=0.5$. The self-adaptive algorithm reduces the $\boldsymbol{P}_c$ and $\boldsymbol{P}_m$ of those individuals whose fitness value are bigger than the average of the current population. It can make the GA operations converge quicker. The $\boldsymbol{P}_c$ and $\boldsymbol{P}_m$ of those individuals whose fitness value are smaller than the average are increased to avoid the local solutions.

□. Fitness Evaluation. The fitness function, used to evaluate the fitness of a solution, is a classifier called SNoW [17]. To train this classifier, we use a new feature pattern to characterize the samples as demonstrated in [2]. And then we train this classifier using the SNoW training procedure. For the details of the SNoW-classifier, please refer to [17]. By the trained classifier, a sample matches a normalized score. Note this score is assigned to the input sample as its fitness.

To train the evaluation function (or classifier) SNoW of each generation, we use the initial population and the solutions of the last generation as positive samples. For negative examples we start with 12,000 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 [15] 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.

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 rotation 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. 3; 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.

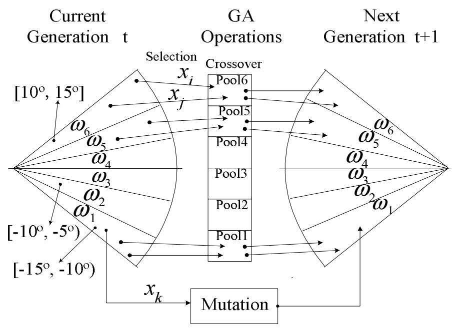


Fig. 3. The process GA operations.

Here, we choose parents randomly to ensure the diversity of the reproduced solutions. As shown in Fig. 3, these selected individuals are put into mating pools, and those individuals within the same sub-set will crossover with the probability P_c as demonstrated in equation (6). 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. Some parents of the current generation mutate with the probability P_m as demonstrated in equation (7). For example, x_k selected from ω_1 is mutated and its child is laid back into ω_1.

After each generation reproduced, we keep 5,000 children at most from the intermediate population and discard the others. Or we keep all of those solutions whose fitness value is bigger than a given threshold θ and we let θ=0.6 according to the experiences. In this scheme, after every 10 generations, the population will be $15{,}000 \times (1+0.3)^{10} = 206{,}787$ 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 only 85,000 solutions. In order to select those 85,000 individuals from 206,787 ones, all of them are sorted in a sequence according to their fitness value and then we pick out 85,000 individuals evenly along this sorted sequence. After these operations, we have 100,000 individuals, which including the 15,000 initial population and their 85,000 children. All the remained children of every 10-generation are checked manually to avoid the classifier assigning a biased fitness value.

After every one generation, we use its solutions 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 is also tested on the validation set. 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. 4.

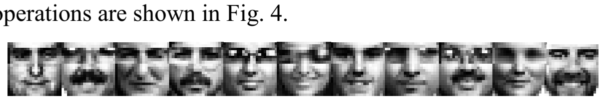


Fig. 4. Some face samples generated by the GA re-sampling.

3. Experiment

3.1. Comparing the solutions performance

Fig. 5 provides the results for the classifier SNoW trained with different database 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 10th generation (GA10) or the 20th, …, or the 60th generation (the same as GA20,…, GA60) as face-sample sets. It means the NoGA has 15,000 face samples, while GA10, GA20,…, or GA60 has 100,000 face samples. For all of these seven cases, the trained classifiers are tested on the validation set.

Form these Receiver Operating Characteristic (ROC) curves in Fig. 5, we can find that the performance of GA10 is better than that of NoGA, and GA20, GA30, GA40 improve the results further respectively. However, the difference between GA50/GA60 and GA40 is very limited. Furthermore, we check these seven trained classifiers by the test set to verify their generalization performance. We find that they obtain almost the same results compared to test on the validation set.

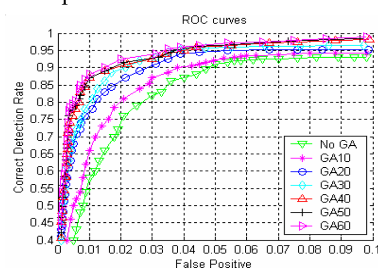


Fig. 5. The ROC curves on the validation set using different generations of GA as training set for a fixed classifier.

We also investigate some possible reasons for the success of GA. First, the random selection scheme ensures the diversity of the reproduced solutions. 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 are very common in our daily life. Third, the mutation can simulate the variations from the lighting and quality conditions in an image to the aging of a person.

3.2. Evaluation of the generated samples

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. Adaboost, a version of the boosting, has been used in face detection and is capable of processing images extremely rapidly while achieving high detection rates [16]. Therefore, we use the Adaboost algorithm to train a classifier. For the details of the Adaboost based classifier, please refer to [3].

To compare the performance improvement on different training sets, we use three different face training sets. The first group consists of the initial population. The second group contains a set of 100,000 face images generated by GA40 ***automatically***. The third group also contains a set of 100,000 face images generated by GA40 ***manually***. 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 10-generation are checked manually. The word "***manually***" means that both each solution of every one generation and all the remained children of every

10-generation are checked manually. The non-face class is initially represented by 10,000 non-face images. Each single classifier is then trained using a bootstrapping approach similar to that described in [15]. The bootstrapping is carried out several times on a set of 8,736 images containing no faces.

The resulting detectors are evaluated on the MIT+CMU frontal face test set which consists of 130 images showing 507 upright faces [12]. The detection performances on this set are compared in Fig. 6. Some detects are shown in Fig.7. From the ROC curves one can find that we get the detection rate of 90.52% and 12 false alarms with the detector trained on the set by GA40 ***automatically***. Viola reported a similar detection capability of 89.7% with 31 false detects (by voting) [16]. 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 [18].

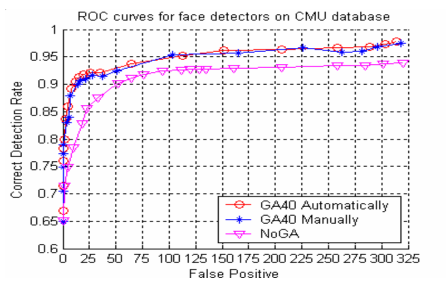
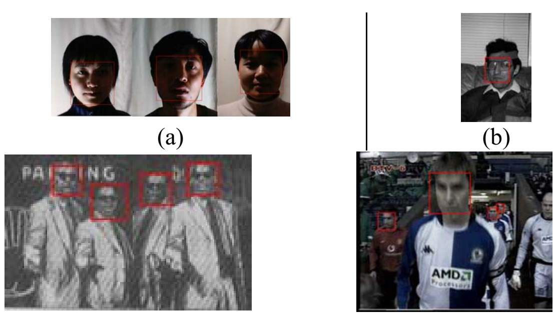


Fig.6. The ROC curves on the MIT+CMU frontal face test set.



(a) (b)

(c) (d)

Fig.7. some results of the detector

4. Conclusions

In this paper, we present a novel method to expand face sample set by applying the self-adaptive 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. It is turned out that 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. It also 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, 2002AA118010 and 2003AA142140), and ISVISION Technologies Co., Ltd.

References

[1] R. Basri and D. Jacobs, "Lambertian reflectance and linear subspaces," *ICCV 2001*, pp. 383 -390.

[2] B. Fröba and A. Ernst. "Fast Frontal-View Face Detection Using a Multi-Path Decision Tree." *In AVBPA '2003*, 2003. pp. 921-928.

[3] 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.

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

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

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

[7] K. C. Lee, J. Ho, and David Kriegman. "Nine points of light: Acquiring subspaces for face recognition under variable lighting." In *CVPR*, 2001.

[8] 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.

[9] 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.

[10] 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.

[11] L. Y. Qing, S. G. Shan, X. L. Chen, "Face Relighting for face recognition under generic illumination." *The 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing.* (Accepted)

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.