Enhancing Human Face Detection by Resampling Examples Through Manifolds

Jie Chen, *Student Member, IEEE*, Ruiping Wang, Shengye Yan, *Student Member, IEEE*, Shiguang Shan, *Member, IEEE*, Xilin Chen, *Member, IEEE*, and Wen Gao, *Senior Member, IEEE*

***Abstract*—As a large-scale database of hundreds of thousands of face images collected from the Internet and digital cameras becomes available, how to utilize it to train a well-performed face detector is a quite challenging problem. In this paper, we propose a method to resample a representative training set from a collected large-scale database to train a robust human face detector. First, in a high-dimensional space, we estimate geodesic distances between pairs of face samples/examples inside the collected face set by isometric feature mapping (Isomap) and then subsample the face set. After that, we embed the face set to a low-dimensional manifold space and obtain the low-dimensional embedding. Subsequently, in the embedding, we interweave the face set based on the weights computed by locally linear embedding (LLE). Furthermore, we resample nonfaces by Isomap and LLE likewise. Using the resulting face and nonface samples, we train an AdaBoost-based face detector and run it on a large database to collect false alarms. We then use the false detections to train a one-class support vector machine (SVM). Combining the AdaBoost and one-class SVM-based face detector, we obtain a stronger detector. The experimental results on the MIT + CMU frontal face test set demonstrated that the proposed method significantly outperforms the other state-of-the-art methods.**

***Index Terms*—AdaBoost, face detection, manifold, resampling, support vector machine (SVM).**

I. Introduction

A. Motivation

OVER the past decades, the problem of human face detection has been thoroughly studied in the computer vision community for its wide potential applications, such as video surveillance, human–computer interaction, human face recognition, face image database management, etc. Given an image, the goal of face detection is to locate the present face/faces in the image and return the location and extent of each face. Recently, much research effort has been directed to learning-based techniques, such as [20], [27], [31], [32], [36]–[39], and [48]. Among them, boosting-based techniques are the new dominant methods, e.g., [9], [25], [26], [44], and [46]. Interested readers are referred to a recent survey by Yang *et al.* [49].

The previous work in learning-based methods has been mainly focused on exploiting robust features, such as [12], [18]–[20], [27], [38], and [44], seeking robust classifiers, such as neural networks [14], [18], [30], [36], [37], [39], [48], boosting classifier performance [25], [26], [44], [46], and using different good algorithms such as support vector machine (SVM) [23], [31], Bayesian classifier [38], etc. However, little attention has been paid to the selection of a suitable training set [45]. In fact, the performance of these learning-based methods highly depends on the training set. In other words, the existing learning-based methods suffer from a common problem: how to collect a limited scale but effective training set? In this research, we focus on obtaining a suitable training set from a large-scale database to train a well-performed face detector yet with any given classifier (e.g., AdaBoost) and feature (e.g., Haar-like features) as mentioned in [44].

As hundreds of thousands of face images are available with the rapidly increased number of web pages and digital cameras, it imposes a new challenge to the research community: how to train a robust classifier based on the collected large-scale database? It is this problem that this paper aims to address. Specifically, we use a manifold to subsample a small subset from the collected face database and then interweave some big holes in the manifold embedding. Likewise, we also resample the nonfaces by the same method. The resampled face and nonface sets are used to train an AdaBoost-based classifier. To further decrease the false alarms of the AdaBoost classifier, we train another one-class SVM classifier based on the false alarms obtained from the trained AdaBoost classifier by running it on a large size of images containing no faces. Combining the AdaBoost and one-class SVM classifier, we demonstrate a robust face detector.

B. Related Work

In this section, we briefly review the previous work on the manifold and its applications in face detection/recognition applications, as well as some resampling methods.

As for the manifold learning techniques, the most prevailing approaches include isometric feature mapping (Isomap) [41], locally linear embedding (LLE) [34], and Laplacian

Manuscript received August 9, 2005; revised June 20, 2006. This work was partially supported by National Natural Science Foundation of China under Contracts 60332010, 60673091, and 60772071, Hi-Tech Research and Development Program of China under Contracts 2006AA01Z122 and 2007AA01Z163, 100 Talents Program of the Chinese Academy of Sciences. This paper was recommended by Associate Editor G. Biswas.

J. Chen is with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: chenjie@jdl.ac.cn).

R. Wang, S. Yan, S. Shan, and X. Chen are with the Institute of Computing Technology and the Key Laboratory of Intelligent Information Processing, Chinese Academy of Sciences, Beijing 100080, China (e-mail: rpwang@jdl.ac.cn; syyan@jdl.ac.cn; sgshan@jdl.ac.cn; xlchen @jdl.ac.cn).

W. Gao was with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China, and the Harbin Institute of Technology, Harbin 150001, China. He is now with the School of Electronics Engineering and Computer Science, Peking University, Beijing 100080, China (e-mail: wgao@jdl.ac.cn).

Digital Object Identifier 10.1109/TSMCA.2007.906575

Eigenmap [2]. Recently, some researchers applied manifold methods to face recognition [1], [12], [15], [30]. Specifically, Fitzgibbon and Zisserman [15] proposed a joint manifold distance to match two sets of images. Fang and Qiu [12] proposed a sample subspace method that projects all the training samples onto each sample to select the sample with the largest accumulated projection. Osadchy *et al.* [30] used a convolutional network to detect faces and estimated their poses. In their method, to estimate the poses, they mapped face images with known poses to the corresponding points on the manifold and nonface images to points far away from the manifold during the training process. Aranjelovic *et al.* [1] proposed a flexible, semiparametric model to learn probability densities for image sets.

As for the resampling techniques, the most widely used methods are bagging [6], [7] and arcing [16], [17]. The bagging method generates several training sets from the original training set and then trains a component classifier from each of those training sets. In contrast, arcing is to adaptively resample and combine the classifiers. The sample weights in the resampling are increased for those most often misclassified. Although we also select a subset from the original training set as bagging, our resampling method is based on the manifold in comparison to the random subsampling of the classical bagging. This means that we subsample the overdense regions and interweave the oversparse regions to obtain a representative subset. Different from arcing, we discard some outliers that might degrade the performance of arcing [7]. Furthermore, we also generate some new virtual samples.

C. Organization of the Paper

The rest of this paper is organized as follows. In Section II, we review Isomap and propose the subsampling method. In Section III, we describe LLE and propose the interweaving method. Experimental results are presented in Section IV, followed by the conclusion in Section V.

II. Subsampling Based on Isomap

In this section, after a brief review of Isomap, we describe how to use it to subsample the training set.

The Isomap algorithm is intuitive, well understood, and produces reasonable mapping results [22], [24], [47]. It captures the geodesic distances between all pairs of data points. It also has a theoretical foundation [3], [10], [50], which has been developed by [5], [21], [33], [35], [40], [42], [43]. Hence, we use Isomap to learn the geodesic distances between pairs of samples and then use these distances to subsample the database.

A. Isomap

Given n data points $\{\boldsymbol{x}_i\}_{i=0}^{n-1}$ in the high-dimensional input space X and the Euclidean distances $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ between all pairs of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, the Isomap algorithm consists of the following three steps [41]. First, a neighborhood graph is constructed. Every two data points ($\boldsymbol{x}_i$ and $\boldsymbol{x}_j$) are connected if the distances $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ are less than a given threshold e or if $\boldsymbol{x}_i$ is one of the K nearest neighbors (KNN) of $\boldsymbol{x}_j$. Second, Isomap computes the shortest path between any two data points. For each pair of nonneighboring data points on the neighborhood graph, Isomap finds the shortest path along the graph with the path bridging from neighbor to neighbor. The length of this path (we call it "the estimated geodesic distance") is an approximation to the true distance between its two end points (we call it "the true geodesic distance"), as measured on the underlying manifold. Finally, the classical multidimensional scaling is used to construct a low-dimensional embedding of the data that best preserves the manifold's estimated intrinsic geometry.

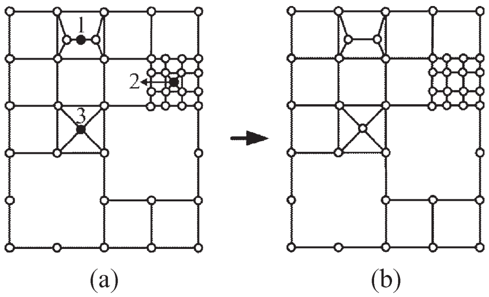


Fig. 1. Illustration of the subsampling scheme based on the estimated geodesic distances.

B. Subsampling

In this section, we discuss the characteristics of the geodesic distance and then demonstrate how to use it to subsample the database.

First, we discuss the special properties of the geodesic distance in comparison to the Euclidean distance. As shown in [41], for two arbitrary points on a nonlinear manifold (for example, in the "Swiss roll" manifold [41]), their Euclidean distance in the high-dimensional input space may not accurately reflect their intrinsic similarity. However, the geodesic distance along the manifold can do it elegantly. In general, the smaller the geodesic distance between two points, the more intrinsic similarity they share. Therefore, we subsample the samples based on the geodesic distances calculated by Isomap.

Subsequently, we show how to subsample the database based on geodesic distances. As discussed above, during the manifold learning, we can obtain the estimated geodesic distances between pairs of samples in a high-dimensional space. In the following stage, these estimated geodesic distances can be used directly to subsample examples by deleting some of them from the database. Moreover, the remaining samples can still keep the data's intrinsic geometric structure. In this way, we obtain a small yet representative subset from a large-scale database.

The proposed scheme is shown in Fig. 1. We sort all the estimated geodesic distances between pairs of samples in an increasing order. We delete a sample if any estimated geodesic distance between this sample and the others is smaller than a given threshold. In our experiment setting, the value of this threshold was decided based on the number of deleted samples. Besides deleting the samples, we also remove those estimated geodesic distances between the deleted samples and the others in the sorted sequence. For example, as shown in Fig. 1, the

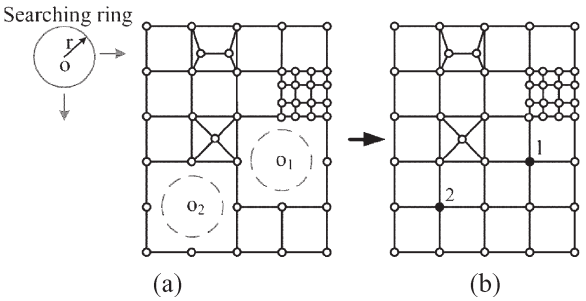


Fig. 2. Illustration of the interweaving scheme in the low-dimensional manifold embedding.

data points 1 and 2 are deleted during subsampling. Data point 3 survives since the estimated geodesic distances between it and the other data points are larger than the given threshold. From Fig. 1(b), one can conclude that the remaining samples can still maintain the data's intrinsic geometric structure.

III. Interviewing Based on LLE

In this section, after a brief review of LLE, we use it to interweave the training set.

The reason to interweave the training set is that the subsampled database by Isomap can still be unbalanced. As shown in Fig. 2, nevertheless, there are some holes in the subsampled subset. How to collect some samples just located in these holes is a difficult problem, particularly in a high-dimensional image space. However, it is much easier to generate some virtual samples to fill in these holes in the manifold embedding. Specifically, to this end, we use the neighbors and their corresponding weights to generate some virtual samples. Meanwhile, the neighbors can be found by the KNN in the low-dimensional manifold embedding, and the weights can be calculated using LLE [34].

We use LLE to compute reconstruction weights since the resulting weights by LLE can best reconstruct samples (i.e., with minimal reconstruction errors) from their neighbors in the low-dimensional embedding. Specifically, different from Isomap, LLE does not attempt to estimate the true geodesics. It recovers a globally nonlinear structure from locally linear fits by the simple assumption that a weighted best fit of each point's neighbors is sufficient to describe the local relationship of the points. Therefore, according to these characteristics of LLE and the essential nonlinear structure of face or nonface manifold, we interweave the database based on the weights computed by LLE in the low-dimensional manifold embedding.

A. LLE

LLE is a method that maps high-dimensional inputs to a low-dimensional and neighbor-preserving embedding [34]. Its procedure is briefly described as follows: 1) assign neighbors (for example, by KNN) to each data point $\boldsymbol{x}_i (i = 0, 1, \ldots, n-1)$, where n is the number of data points; 2) compute the weights $\boldsymbol{\omega}_{ij} (j = 0, 1, \ldots, K-1)$ that best linearly reconstruct $\boldsymbol{x}_i$ from its neighbors; and 3) compute vectors $\boldsymbol{y}_i$ best reconstructed by $\boldsymbol{\omega}_{ij}$ in the low-dimensional embedding.

B. Interweaving

In this section, after discussing the properties of LLE, we show how to find the oversparse regions in the manifold embedding and generate virtual samples.

LLE is an unsupervised learning algorithm. It computes low-dimensional and neighborhood-preserving embeddings of high-dimensional inputs. Unlike the clustering methods for local dimensionality reduction, LLE maps its inputs into a single global coordinate system of lower dimensionality and avoids the local minima in its optimizations. Through exploiting the local symmetries of linear reconstructions, LLE is able to learn the global structure of nonlinear manifolds, e.g., a face manifold.

In addition, the weights by LLE share the advantages that 1) they can be computed easily by methods in linear algebra; 2) they can best linearly reconstruct vectors (i.e., with the minimal reconstruction errors) from its neighbors in the low-dimensional embedding; and 3) they are symmetric: for any particular data point, the weights are invariant to rotations, rescalings, and translations of the data point and its neighbors. Note that, in our experiments, we compute the weights in the low-dimensional embedding and reconstruct the high-dimensional vectors. The process is reversed as in LLE.

After computing the weights by LLE, the procedure of generating virtual samples to interweave the oversparse regions in low-dimensional embedding consists of two steps: the first is to find the sparse regions in the embedding, and the second is to reconstruct the virtual samples by their neighbors and the corresponding weights.

First of all, we need to find the sparse regions in the manifold embedding. For example, as shown in Fig. 2, due to the unbalance of the database, there are still some "holes," such as the position of circles O_1 and O_2, in the low-dimensional embedding after subsampling. How to search these "holes" in the embedding? In our case, first, we calculate the median d_m of all of the Euclidean distances between pairs of points $\boldsymbol{y}_i (i = 0, 1, \ldots, n-1)$ and its K neighbors $\boldsymbol{y}_j (j = 0, 1, \ldots, K-1)$. The value d_m is used as the radius of "searching ring" as shown in Fig. 2. Second, moving the searching ring by a given experiential step-length in the embedding, we find several holes (herein, when the searching ring runs over the embedding, a hole is hit if no data point is located within it). The centers of these holes, such as points 1 and 2, are the places where we generate virtual samples. Note that we run the searching ring along the embedding in its 2-D project for convenient computation. We have repeated this operation in higher dimension projections (e.g., 3-D, 4-D, etc.) and find that the performance difference among the resulting classifiers is marginal.

Having found the holes in the embedding, the next step is to generate some virtual examples to fill in these holes. In our case, after learning the embedding, we fill in these holes with some virtual examples. The process to generate a virtual sample is as follows.

1) Find the K neighbors $\{\boldsymbol{y}_{p1}, \boldsymbol{y}_{p2}, \ldots, \boldsymbol{y}_{pK}\}$ of a virtual example $\boldsymbol{y}_p^{\mathrm{VE}} (p = 0, \ldots, m-1)$ in the low-dimensional embedding, where a virtual example corresponds to the

TABLE I
ORIGINAL SAMPLES VERSUS VIRTUAL SAMPLES

Original samples	
Virtual samples	

TABLE II
VIRTUAL SAMPLES AND THEIR CORRESPONDING NEIGHBORS

a virtual sample of a Caucasian face						
neighbors						
weights	0.4027	0.3497	0.3130	0.2815	-0.0313	-0.3156
a virtual sample of an African American						
neighbors						
weights	1.8000	1.4710	0.3470	-0.3339	-0.7530	-1.5310
a virtual female sample						
neighbors						
weights	0.4092	0.2897	0.1338	0.1327	0.0638	-0.0293

center of a found hole; the superscript "VE" denotes virtual example, and m is the number of virtual examples.

2) Compute the weights $\omega_{pj}(j = 0, 1, \ldots, K - 1)$ by LLE for the best linear reconstruction of y_p^{VE}.
3) Reconstruct the virtual example $\boldsymbol{x}_p^{\mathrm{VE}}$ by ω_{pj} and its K neighbors in the high-dimensional input space as follows: $\boldsymbol{x}_p^{\mathrm{VE}} = \sum_{j=1}^{K} \omega_{pj}\boldsymbol{x}_{pj}$, where $\{\boldsymbol{x}_{p1}, \boldsymbol{x}_{p2}, \ldots, \boldsymbol{x}_{pK}\}$ are the K neighbors of the virtual example $\boldsymbol{x}_p^{\mathrm{VE}}$, and $\boldsymbol{x}_p^{\mathrm{VE}}$ is the data point in the high-dimensional input space corresponding to $\boldsymbol{y}_p^{\mathrm{VE}}$, likewise, $\{\boldsymbol{x}_{p1}, \boldsymbol{x}_{p2}, \ldots, \boldsymbol{x}_{pK}\}$ corresponding to $\{\boldsymbol{y}_{p1}, \boldsymbol{y}_{p2}, \ldots, \boldsymbol{y}_{pK}\}$.

Some virtual face samples are shown in Table I, and some virtual samples and their neighbors are shown in Table II. From Table I, one can conclude that these virtual samples look very much like real faces. From Table II, one can conclude that a virtual sample preserves the appearance information of the face classes that generate the sample. For example, a virtual sample generated by a group of Caucasian faces looks like a Caucasian face, while a virtual sample generated by a group of African-Americans looks like an African-American. A female face will generate a virtual female face.

However, as shown in Table II, some weights are negative, which might result in minus pixel values. To avoid this case, we normalize the pixel values of the virtual samples to [0, 255] again.

IV. EXPERIMENTS

In this section, we carried out two groups of experiments to verify the proposed method. One group was performed to compare the classifiers trained by the original training set and resampled ones. The other used the resampled training set to train a face detector and then test it on the MIT + CMU frontal face test set.

A. Experiments on the MIT Face Database

In this group of experiments, we verified the effectiveness of the proposed method and compared the performances of the trained classifiers by the originally collected training set and that by several resampled ones. Specifically, the training sets were composed of the following five cases: 1) the originally collected training set; 2) the subsampled ones by Isomap at different ratios; 3) the subsampled ones by random; 4) the training sets subsampled by Isomap and then interweaved by LLE; and 5) the training sets subsampled by Isomap and then interweaved by PCA.

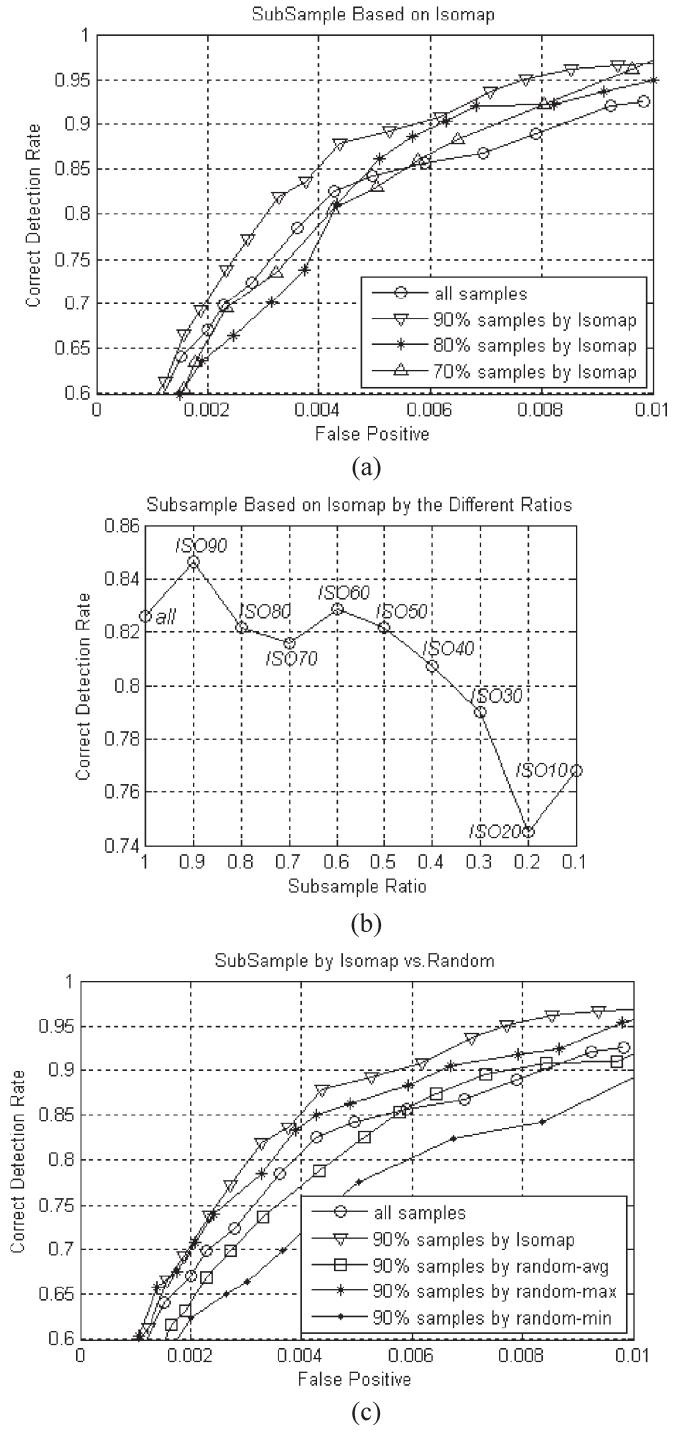
The data set was from the Massachusetts Institute of Technology (MIT) Center for Biological & Computational Learning (CBCL) webpage (http://cbcl.mit.edu/software-datasets/FaceData2.html), which consists of a training set of 6977 samples and a test set of 24 045 samples. Meanwhile, the training set is composed of 2429 faces and 4548 nonfaces. We call it the original training set. The test set is composed of 472 faces and 23 573 nonfaces. All of these samples are in grayscale mode and have been normalized to 19 × 19.

1) Face Example Subsampling: In this section, we compare the trained classifiers based on the following training sets: the original training set, the training sets with the face examples subsampled by Isomap at different ratios, and the training sets with the face examples subsampled by random.

To subsample face examples in the original training set, we first learned the manifold of these faces. Specifically, we let $K = 6$ to experientially learn the manifold of 2429 faces in the original training set. By the Isomap, we obtained estimated geodesic distances between pairs in the high-dimensional space. Afterwards, these distances were used directly to subsample the faces by deleting some samples. Note that all of these samples were performed by histogram equalization before manifold learning. This is because, to train a classifier, all samples are needed to perform histogram equalization. That is, it normalizes the histogram of face samples and makes faces more compact in the image space. Therefore, it is widely used in face detection.

Having learned the globally nonlinear structure of faces by Isomap, we then subsampled the face samples. In order to verify the effects of different subsampling ratios on the trained classifiers, we subsampled the faces by 90%, 80%, 70%, ..., and 10% as discussed in Section II-B. We call the subsampled sets as ISO90, ISO80, ISO70, etc. Meanwhile, ISO90 means we preserve 90% of the samples, and the same meaning applies for ISO80, ISO70, etc. Note that ISO70 is a subset of ISO80, ISO80 is a subset of ISO90, and so on.

The nine subsampled face sets (ISO90 and the others) together with the nonface set were used to train nine classifiers based on the AdaBoost (for the details about how to train a classifier based on AdaBoost, please refer to [44]). Likewise, all faces in the original training set were also used to train an AdaBoost-based classifier with the same nonfaces. The ten



(a)

(b)

(c)

Fig. 3. ROC curves among the trained classifiers on the MIT test set. (a) Comparison of the trained classifiers based on the face sets subsampled by Isomap and the original set. (b) Comparison of the equal error rates of the trained classifiers based on the different subsampling ratios by Isomap. (c) Comparison between the trained classifiers based on the face sets subsampled by Isomap and by random.

resulting classifiers (one by all faces and nine by subsampled faces) were then tested on the test set. Three receiver operating characteristic (ROC) curves for the three classifiers trained on ISO90, ISO80, and ISO70 are shown in Fig. 3(a). The equal error rates (i.e., the intersection of false rejection rate and false acceptance rate) of the ten classifiers are shown in Fig. 3(b).

From Fig. 3(a) and (b), one can conclude that five of their detectors (i.e., based on ISO90, ISO80, ..., and ISO50) achieved a comparable performance in comparison to the detector based on all faces. It demonstrates that it is reasonable to subsample the training set by Isomap. Furthermore, the detector trained by ISO90 is the best one, and it improved distinctly the performance of the trained detector compared with the detector by all faces. We believe that the evenly distributed samples and no outliers (we discard 30 outliers by Isomap) contributed to the good results. We will discuss these reasons in detail in Section IV-B1.

Herein, a key issue is to determine threshold values for subsampling. In general, it depends on the size of the training set and the target distribution. First, test errors usually decrease in a typical monotonic power-law function with the increase of the training set size [11]. That is, if the size of the training set is smaller, we only prune a smaller ratio and preserve more samples. Indeed, removing the outliers by Isomap for a small training set obviously improves the system performance. On the contrary, if the size of the training set becomes larger, we prune a larger ratio of the samples. Second, if the target distribution is very simple (e.g., a Gaussian distribution) and/or the target class is easy to be classified from the other class (e.g., to classify handwritings 1 from 8), one can subsample the training set by a larger ratio and vice versa.

For example, for a small training set (2429 faces in this section), we preserve 90% of the faces when subsampling. In contrast, for a large training set (1 500 000 in Section IV-B), we preserve only 14 000 face samples (less than 0.1%) since 14 000 faces are enough to train a well-performed classifier as demonstrated by experiments.

However, random subsampling does not work as well as the Isomap subsampling. We randomly subsample faces from the original training set for 1000 times and obtain 1000 subsets. Each subset has the same number of samples as ISO90. Likewise, all of these randomly subsampled subsets together with nonfaces in the original training set were used to train the AdaBoost-based classifiers. The resulting classifiers were also tested on the test set. Some ROC curves are shown in Fig. 3(c). In this figure, we plot the ROC curves of the detectors based on all faces, ISO90, and two randomly subsampled subsets (i.e., best and worst cases) together with the average performance for random. Herein, the curve "90% samples by random-avg" denotes the average performance of these 1000 subsets subsampled randomly; the curves "90% samples by random-min" and "90% samples by random-max" represent the worst and best results of these 1000 random-subsampling cases.

From these ROC curves in Fig. 3(c), one can conclude that the detector based on ISO90 is still the best, and the results based on random subsampling are very unstable. We also conclude that the evenly distributed examples and no outliers contribute to the results. We will discuss these reasons in details in Section IV-B1.

2) Face Example Interweaving: In this section, we study the effects of virtual samples on the trained classifiers.

First, we discuss the effects of different numbers of virtual samples by LLE. Specifically, after the subsampling by Isomap, we interweaved the manifold embedding as discussed

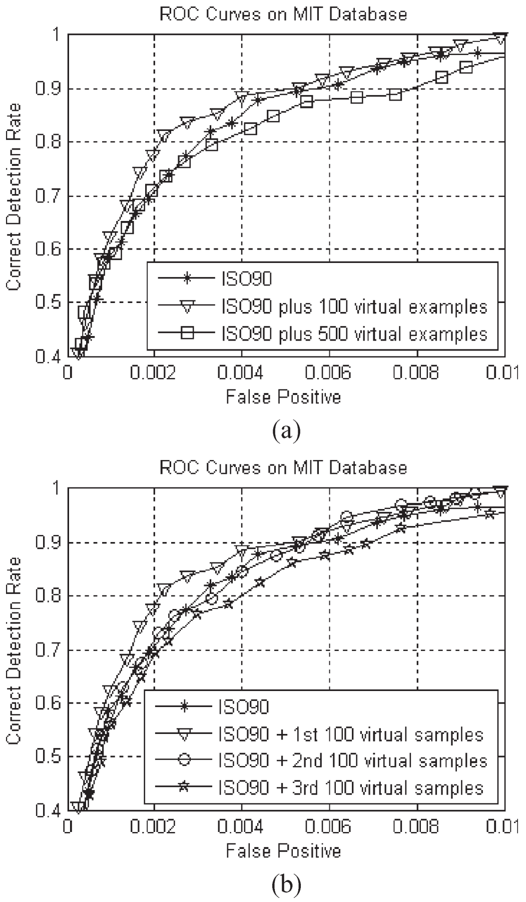


Fig. 4. ROC curves on the virtual samples by LLE. (a) Adding different numbers of virtual examples. (b) Changing radius of a searching ring.

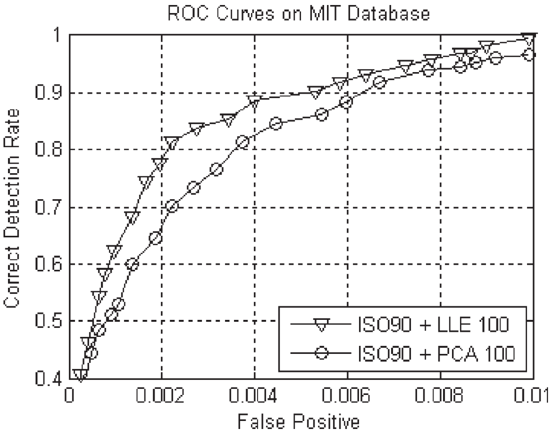


Fig. 5. ROC curves on different interweaving methods.

in Section III-B. As shown in Fig. 4(a), we added the different numbers of virtual examples in the set ISO90 (i.e., 100 or 500 examples, respectively). One can conclude that adding a few examples is valuable for training a detector. When the number is up to 500, it is not as good as 100 examples. Moreover, we test a detector based on ISO90 with 300 virtual examples. We find that the performance of the detector by adding 300 virtual examples in ISO90 is better than that of adding 500, but inferior to that of adding 100 examples.

Second, we gained an insight into the effects of the sparse or dense degree of virtual samples on the resulting classifiers. As shown in Fig. 4(b), we changed the radius of the searching ring. The first 100 virtual examples were generated by moving the searching ring with the radius equal to d_m. The second 100 virtual examples were generated by the ring radius $1.1 \times d_m$ and the third 100 virtual examples by $1.2 \times d_m$. One can conclude that, when the radius is equal to d_m, the added 100 examples are the most valuable for training a detector. We also find that, when the radius is equal to $0.9 \times d_m$ or $0.8 \times d_m$, the performance improvement is not as valuable as that by the radius equal to d_m.

These two groups of experiments indicate that the training set should distribute reasonably instead of being overdense or oversparse. For example, when the radius of the searching ring becomes larger than d_m, the local embedding patches (where the virtual examples are located) become sparser. In contrast, when the radius becomes smaller than d_m, the local embedding patches become denser.

Herein, the key issue is to determine how many virtual samples should be generated. In general, it depends on two factors. One is the size of the collected data set, and the other is the distribution of the data set. Specifically, on one side, if the size of the collected data set is larger, we usually generate more virtual samples and vice versa. On the other side, if the data set distributes relatively sparsely, we generate more samples in comparison to the case that the data set distributes densely. In our experiment, the number of the virtual samples is determined experimentally. An experiential formula is $m = \lambda n$, where m is the number of the virtual samples, λ is a constant within [0.02, 0.1], and n is the size of the data set.

3) LLE and PCA: In this section, we compare two different interweaving methods—LLE and principal components analysis (PCA).

PCA is a useful statistical technique. It has been successfully applied in many fields, such as face recognition and image compression. Moreover, it is a common technique for finding patterns in data of high dimension. In [45], we reconstructed virtual examples by performing PCA locally among the reconstructed examples and their neighbors.

As shown in Fig. 5, we added 100 virtual examples in the set ISO90 based on different methods. The resulting face sets (ISO90 + LLE100, ISO90 + PCA100) together with the nonface set were used to train two AdaBoost-based classifiers. Likewise, the two trained classifiers were tested on the MIT test set. One can conclude that the classifier performance based on ISO90 + LLE100 is better than that of ISO90 + PCA100. We believe that there are two reasons. First, it is due to the essential nonlinear structure of faces, on which LLE can work but PCA cannot. Second, the weights computed by LLE share some advantages (please refer to Section III) in comparison to PCA.

B. Experiments on the Real Applications

In this group of experiments, first, we resampled the collected large-scale face database. Second, we resampled the collected large-scale nonface database. Third, we modeled the false alarms by one-class SVM. Finally, we compared the resulting detector with the state-of-the-art detectors.

The original face set consisted of 100 000 samples. They were collected from the Internet, video, and digital cameras and covered wide variations in pose, facial expression, and also in different lighting conditions. To make the detection method robust to affine transformation, the images were often rotated, translated, and scaled [19]. After these preprocessing, we obtained 1 500 000 face examples, and we called it whole set. Subsequently, we subsampled the whole set. The first group was subsampled by Isomap. It included 14 000 face examples, and we called it ISO14000. The second group was randomly subsampled from the whole set for ten times. Each randomly subsampled subset was also composed of 14 000 face examples.

The nonface database consisted of 311 085 images containing no faces. They are from the Corel database (www.corel.com), LabelMe database (http://www.csail.mit.edu/~brussell/research/LabelMe/intro.html) and other websites. First, from this nonface database, we randomly picked out 11 085 images to train the AdaBoost-based classifier with only faces resampling; second, we used 100 000 images for the nonfaces resampling during the training of the AdaBoost-based classifier. Third, we used the remaining 200 000 images to collect false alarms and then modeled the obtained false detections by one-class SVM.

The test set in this section was MIT + CMU frontal face test set, which consists of 130 images showing 507 upright faces [36].

1) Face Resampling: In this section, we only resampled faces to verify its effects on the large-scale face database. The nonface examples were from 11 085 images.

In order to subsample the face set, we need to learn its manifold. However, it is hard to learn the manifold from 1 500 000 examples by Isomap because it needs to calculate the eigenvalues and eigenvectors of a 1 500 000 × 1 500 000 matrix. To avoid this problem, as shown in Fig. 6, we divided the whole set into 500 subsets, and each subset has 3000 examples. We got 1000 examples by Isomap from each subset and then incorporated every three subsampled sets into one new subset. With the same procedure, we obtained a final subset by Isomap with totally 14 000 examples after incorporating all subsampled examples into one set.

To avoid destroying the intrinsic structure of the data manifold when the whole set is divided into 500 subsets, we divided samples with similar variations into the same subset. That is, the examples with the similar pose fall into the same/neighbor subsets and the same for the variations of facial expression or lighting condition. Likewise, this criterion is applied to incorporate the subsampled subsets.

Besides the subsampling by Isomap, we also randomly subsampled ten subsets from the whole set. Each subset consisted of 14 000 examples as well.

For the nonface examples, they were initially represented by 14 000 nonface samples. Each layer in the AdaBoost cascade was then trained by using a bootstrap approach [39]. In this way, we increased the number of negative examples. The bootstrap was carried out several times on a set of 11 085 images.

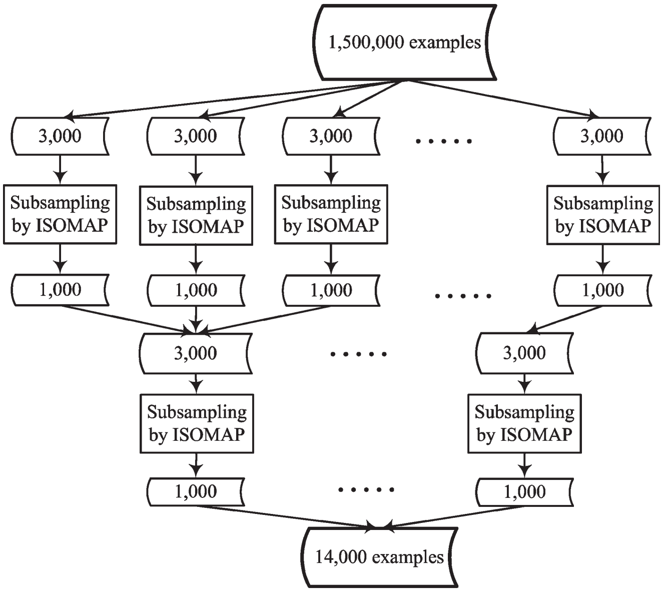
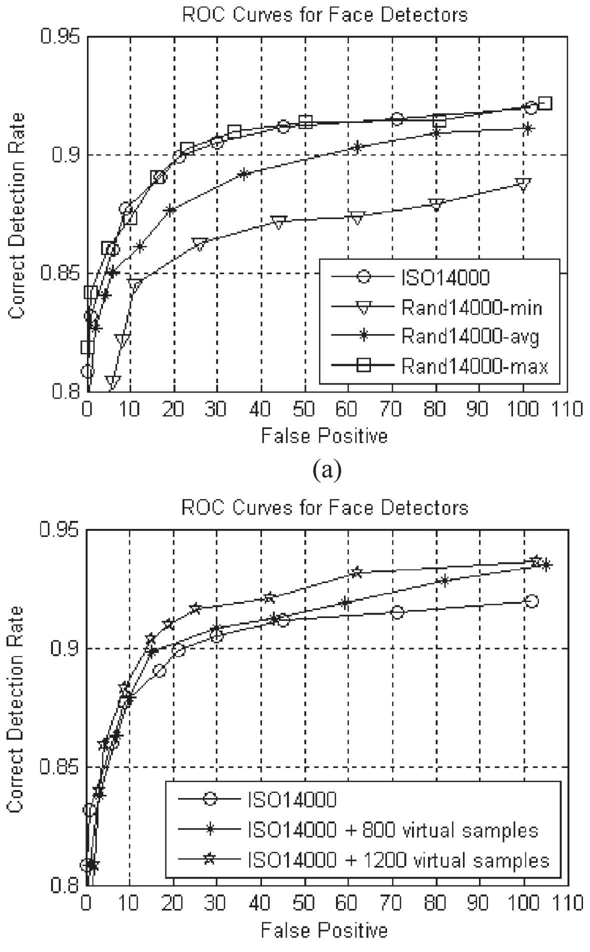


Fig. 6. Subsampling procedure by Isomap to obtain 14 000 examples from 1 500 000 ones.

The subsampled face sets by Isomap and by random were both used to train the AdaBoost-based classifiers. Note that we have trained 11 detectors, one using the face set subsampled by Isomap and ten by random. The trained detectors were evaluated on MIT + CMU frontal face test set. The results on this set are compared in Fig. 7(a). Herein, the curve "ISO14000" denotes the results of the detector trained on ISO14000, the curve "Rand14000-avg" denotes the average performance of these ten subsets subsampled randomly, and the curves "Rand14000-min" and "Rand14000-max" represent the worst and best results of these ten random-subsampling cases.

From these ROC curves in Fig. 7(a), one can conclude that the detector based on ISO14000 is much better than that of Rand14000-avg, particularly for the case of Rand14000-min. Moreover, the detector on ISO14000 achieves the comparable performance to Rand14000-max. We believe that there are some possible reasons. First, the examples subsampled by Isomap distribute evenly in the example space and have no examples congregating compared with the whole set. Therefore, subsampling by Isomap adjusts the positions of the weak classifiers and offset/deflect them to more reasonable direction during the detector training based on AdaBoost. Herein, a weak classifier means that a classifier is not needed to be the best classification function to classify the training data well, i.e., for any given problem the best perception may only correctly classify the training data 51% of the time. For more, please refer to [6], [7], [16], [17], and [44]. Second, the outliers in the whole set have been discarded during the manifold learning [41] (we discard 1638 outliers by Isomap.). This is because the boundary between faces and nonfaces becomes clear by discarding the outliers. In turn, it alleviates the overfitting brought forth by the outliers and improves the generalization ability of the trained classifier.

Similarly, from the ROC curves in Fig. 7(a), one can observe that the results based on the random subsampling are very



(a)

(b)

Fig. 7. ROC curves for the trained detectors. (a) Results of the training sets by Isomap and by random. (b) Results of the training sets by adding virtual samples.

unstable. The performance difference between Rand14000-max and Rand14000-min is almost 6%, particularly for the low false alarms. It may be that different random subsamples result in different distributions and different outliers of the subsampled subsets, which, in turn, leads to the instability of the trained detectors. That is also the reason why we obtain the different results, even when using the same number of examples, the same classifier (AdaBoost or SVM based), the same parameters for the selected classifier, and the same test set, but different training samples by the random subsampling from the face space.

One might note the difference between the results by Isomap and by random in Figs. 3(c) and 7(a). Specifically, the difference between the Isomap and random in Fig. 3(c) is more obvious than that in Fig. 7(a). We believe that the outliers degrade the performance of the trained classifiers on a small size of training set more than that of a larger size one. It is consistent with the conclusion from [7].

After subsampling the faces by Isomap, we obtained the training set ISO14000 as mentioned above. Subsequently, we added some virtual examples (800 and 1200, respectively). As shown in Fig. 7(b), both of the two detectors trained by ISO14000 together with the added virtual examples outperform the detector only by ISO14000. That is, the added virtual examples improve the detector's performance further. We believe that the generated samples make the distributions of the resulting face set approach to the target distribution much better.

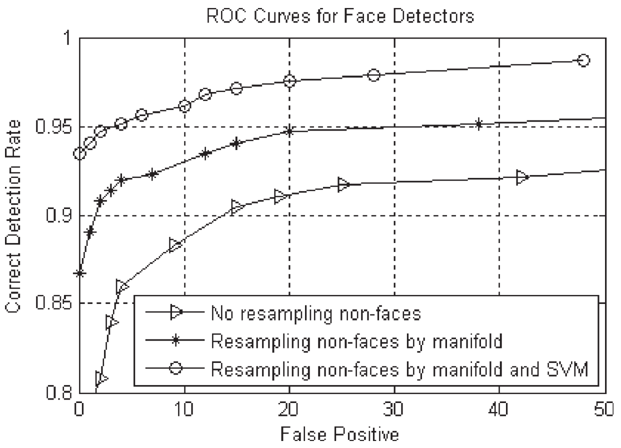


Fig. 8. Results of nonface resampling by manifold modeled by a one-class SVM.

2) Nonface Resampling: In face detection, the nonfaces space is unbounded, and they can play an important role. Therefore, the system performance sometimes highly depends on the choice of nonfaces. In this section, in order to reduce the false alarms of the trained face detector, we also resample the nonfaces by the proposed method.

Besides resampling faces, we also used the proposed method to resample nonfaces. That is, we used Isomap to subsample the collected nonfaces and then applied LLE to interweave them. However, different from faces resampling, we resampled nonfaces during the training process of the AdaBoost-based face detector. Specifically, for each layer in training a cascade of classifiers, we resampled the nonfaces that were bootstrapped from the images containing no faces. In our experiments, we collected 100 000 misclassified nonfaces by the former layer, and then, we subsampled them to obtain one subset consisting of 14 000 examples. Afterwards, we interweaved the subset and also generated about 1200 virtual nonfaces for each layer.

The performance comparison with/without nonfaces resampling is shown in Fig. 8. It can be observed that the nonfaces resampling by manifold can improve the system performance nearly by 3%–4%, particularly for the low false alarms by 6%–8%. From this observation, we believe that the well-distributed nonfaces contribute to these improvements. Furthermore, resampling on a larger data set containing no faces obtains more representative nonfaces to train a face detector, which helps to clarify the boundary between faces and nonfaces.

3) One-Class SVM for False Alarms: In this section, we use one-class SVM to model false alarms to reduce them further in face detection.

First, we collected false alarms. To this end, we ran the trained detector in Section IV-B2 on the collected nonface database (200 000 images containing no faces). The obtained false alarms (denoted as S_{fa}) were then modeled by one-class ν-SVM using radial basis function (RBF) as the kernel function [8].

We first used only the set S_{fa} to learn a one-class SVM-based classifier. Subsequently, we ran this classifier on the face set ($\mathrm{ISO14000} + 1200$) to collect some false negatives (denoted as S_{fn}). Afterward, we combined S_{fa} and S_{fn} to train a final one-class SVM-based detector. In our case, we obtained 10 592 false alarms and 3681 false negatives; the parameter ν, i.e., fraction of errors, is equal to 0.01, and the parameter σ, i.e., width of the RBF kernel, is equal to 1200.

The final face detector consisted of two subclassifiers—one was the AdaBoost-based classifier and the other was the one-class SVM-based classifier. Note that only those subwindows that pass the AdaBoost-based classifier were then used as inputs of the one-class SVM-based classifier. By this means, the false alarms can be reduced sharply. As shown in Fig. 8, we achieved the detection rate of 93.5% without false alarms. As far as we know, it is the best result with no false alarm on the MIT + CMU frontal face test set. The detection rates were also improved significantly by 3%–6%, particularly for the low false detections.

Similar to the work on face pattern modeling proposed by Jin *et al.* [23], during the preparation of training set, our target object is also the face samples. However, different from their work, we subsample faces and generate some new virtual faces. Furthermore, during the classifier training, we also resample nonfaces to help clarify the boundary between faces and nonfaces.

Although Jin *et al.* have also bootstrapped their detector on the faces, the performance of our system is superior to theirs—they have tested on a subset, including 38 images showing 164 faces, of MIT + CMU test set. They achieved 90.24% detection rate with 24 false alarms. We believe that, as shown in [28] and [29], such one-class approaches are typically applied in novelty detection. In these type of applications, it is often the case that negative data (e.g., only a few measurements of a power plant that is out of the "normal" range) are very rare. Therefore, the absence of negative information leads to the case that one should not expect so good results as the negative information is available. Furthermore, in face detection, although nonfaces are hard to be modeled, we can collect many nonfaces easily. Thus, if one trains a face detector both by faces and nonfaces, he/she always obtains a classifier with a better performance.

4) Comparison With Others: In this section, we compare our method to the state-of-the-art methods.

As shown in Fig. 9, we address the performance comparison between our system and some existing face detection algorithms on MIT + CMU test set, such as Féraud *et al.* [14], Garcia and Delakis [18], Li and Zhang [25], Rowley *et al.* [36], Schneiderman and Kanade [38], and Viola and Jones [44].

From the ROC curves in Fig. 9, one can observe that our method compares favorably with the others, particularly for the low false alarms. When the false alarms are about ten, our system outperforms that of Viola and Jones (without voting) [44], who also trained an AdaBoost-based face detector, by nearly 20%.

Some face detectors based on the manifold are also developed, such as Osadchy *et al.* [30] and Fang and Qiu [12]. In Tables III and IV, we list the detection rates for various numbers of false alarms for our system, as well as for [30] and [12]. It can be observed that our method outperforms others, particularly that our system has very low numbers of false alarms. It shows that our approach separates face and nonface space in a robust and balanced way.

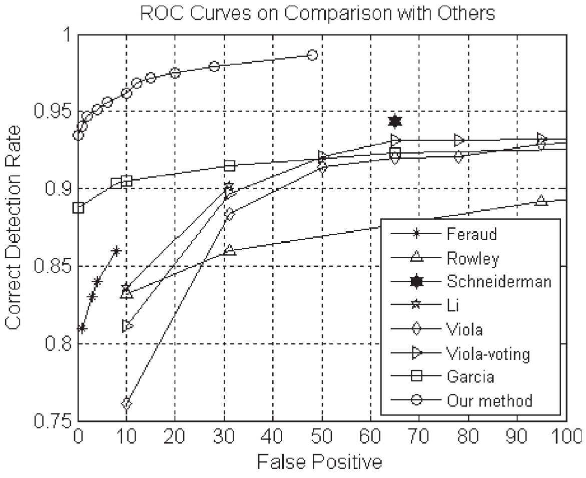


Fig. 9. ROC curves comparison of our method and others tested on the MIT + CMU frontal face test set.

TABLE III
COMPARISON WITH OSADCHY–MILLER–LECUN [30]

Data set	MIT +CMU				
False positives per image	0	0.07	0.37	0.50	1.28
Our detector	93.5%	96.1%	98.7%	N/A	N/A
Osadchy et al. [30]	N/A	N/A	N/A	83%	88%

TABLE IV
COMPARISON WITH FANG–QIU [12]

Data set: MIT +CMU (set A, B, C)				
Algorithm	Faces(Images)	Faces Correctly Detected	Correct Detection Rate	False Positive Rate
Our detector	507(130)	474	93.5	0
		487	96.1	2.40E-7
Fang –Qiu[12]	507(130)	448	88.3%	1.26E-5

However, different criteria (e.g., training time, number of training samples involved, cropping training set with different subjective criteria, execution time, and the number of scanned windows during detection, etc.) 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 [49]. Some results of our detector are shown in Fig. 10.

We conclude that resampling on both faces and nonfaces by the proposed method improves significantly the performance of the finally trained classifier. In detail, first, for face resampling, subsampling by Isomap improves the system performance by about 3%, and interweaving by LLE furthers by about 2%; second, for nonface resampling, subsampling and interweaving gain nearly 5%; third, modeling false alarms by one-class SVM-based classifier increase the resulting system by about

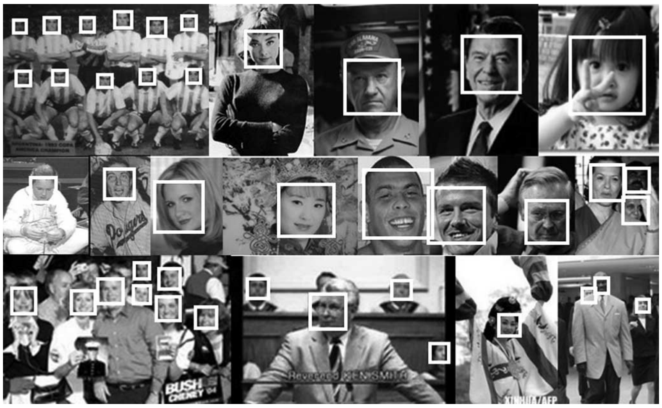


Fig. 10. Some results of our trained detector.

3%. Particularly for the low false alarms, we have increased the detection rate totally by nearly 20%.

C. Discussion

In this section, we discuss the influence of sample distributions and outliers on the trained classifiers. As mentioned in Sections IV-A and B, the experimental results indicate that the evenly distributed examples and no outliers contribute to the improvement on the performance of the trained detectors. How can these two factors work?

The first factor is that different subsampling methods result in different sample distributions. The samples subsampled based on manifold distribute evenly in the sample space without congregation in comparison to the whole set and randomly subsampled subsets. For example, as shown in Fig. 11, subpanel (a) denotes a weak classifier l_{o} between the boundary of face and nonface sets collected originally, subpanel (b) denotes a weak classifier l_{r} between the boundary of face and nonface subsets randomly subsampled from the whole sets, and subpanel (c) denotes a weak classifier l_{m} between the boundary of face and nonface subsets subsampled based on the manifold. Because of the random subsampling, face samples in the cluster near the nonface set could be neglected innocently since one does not know the distribution of the face samples in the data space. Therefore, the weak classifier l_{r} might be far away from l_{o}. However, subsampling based on the manifold can avoid this kind of nescience, as shown in Fig. 11(c). It is because subsampling based on the manifold can still maintain the data's intrinsic geometric structure.

The second factor is that the outliers in the whole set can deteriorate its performance. However, these outliers are discarded during the manifold learning. In turn, the subsampled training set by manifold improves the performance of the trained classifier. For example, as shown in Fig. 12, subpanel (a) denotes two weak classifier l_1 and l_2 between the boundary of face and nonface sets collected originally and subpanel (b) denotes a weak classifier l between the boundary of face and nonface subsets after discarding the outliers. On account of the outliers in the originally collected face set, we have to use two weak classifiers to partition the faces and nonfaces to guarantee the detection rates and false alarms of the trained detector. However, after discarding the outliers during the manifold learning, the boundary becomes much clearer. A weak classifier l can carry out the classification as well. In turn, it can improve the generalization ability of the trained face detector.

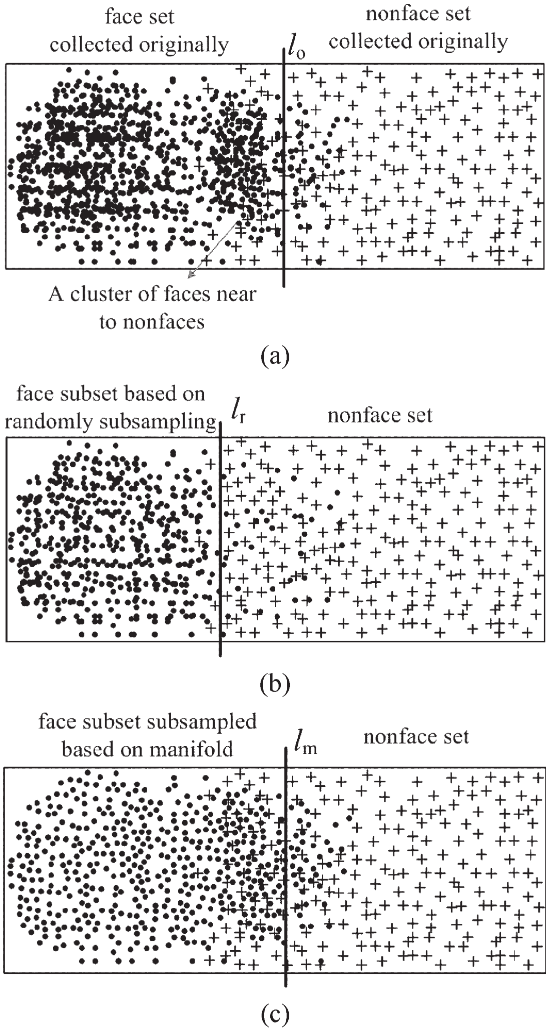


Fig. 11. Illustration of the influences of different subsampling methods on weak classifiers during detector training based on AdaBoost.

V. Conclusion

In this paper, we have presented a novel manifold-based method to obtain a relatively small but effective training set from a collected large-scale face/nonface database. We subsample the database based on the estimated geodesic distances between pairs of samples by Isomap. Afterward, we interweave the training set based on the weights computed by LLE in the low-dimensional manifold embedding. Subsequently, we train an AdaBoost-based face detector by the resampled training set and then run the detector on a large database to collect false alarms. These false alarms are then modeled by a one-class SVM-based classifier. A final face detector is composed of the AdaBoost and one-class SVM-based classifiers. Compared with the detectors trained by the random-subsampling sample sets, the detector trained by the proposed method is more stable and achieves better performance compared with the other popular methods. Moreover,

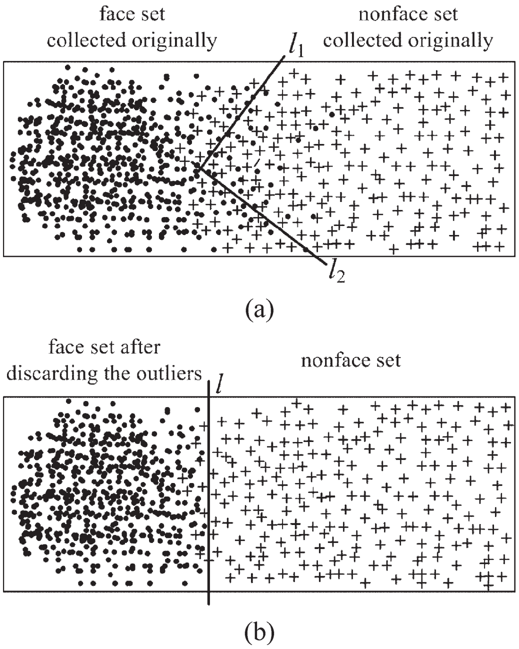

face set collected originally
nonface set collected originally
l_1
l_2
(a)
face set after discarding the outliers
nonface set
l
(b)

Fig. 12. Illustration of influences of outliers on weak classifiers during detector training based on AdaBoost.

the experimental results show that, in order to improve the performance of a face detector, how to collect a suitable training set is as much important as how to design a suitable classifier. Although the proposed method is used to resample faces and nonfaces in this paper, it can be applied to resample other database (e.g., cars [38], handwritings [4], image patches [13], and so on). How to use the proposed method to resample other database is our future work.

Acknowledgment

The authors would like to thank the Associate Editor and the anonymous reviewers, whose comments helped to improve the paper. In particular, the authors are indebted to one of the reviewers for his suggestion on nonfaces resampling reported in Sections IV-B2 and B3 by which the performance of our trained system was improved significantly. The authors would also like to thank Dr. J. Yang from CMU for his help on improving the manuscript.

References

[1] O. Aranjelovic, G. Shakhnarovich, J. Fisher, R. Cippola, and T. Darrell, “Face recognition with image sets using manifold density divergence,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2005, pp. 581–588.

[2] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” in *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press, 2002, pp. 585–591.

[3] M. Bernstein, V. de Silva, J. Langford, and J. Tenenbaum, “Graph approximations to geodesics on embedded manifolds,” Stanford Univ., Stanford, CA, 2000. Tech. Rep.

[4] A. Biem, “Minimum classification error training for online handwriting recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1041–1051, Jul. 2006.

[5] M. Brand, “Charting a manifold,” in *Advances in Neural Information Processing Systems 15*. Cambridge, MA: MIT Press, 2003, pp. 961–968.

[6] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[7] L. Breiman, “Arcing classifiers,” *Ann. Stat.*, vol. 26, no. 3, pp. 801–849, 1998.

[8] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, 2001. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[9] J. Chen, X. Chen, and W. Gao, “Expand training set for face detection by GA re-sampling,” in *Proc. 6th IEEE Int. Conf. Autom. Face Gesture Recog.*, 2004, pp. 73–79.

[10] D. L. Donoho and C. Grimes, “When does Isomap recover natural parameterization of families of articulated images?” Stanford Univ., Stanford, CA, Tech. Rep. 2002-27, 2002.

[11] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ: Wiley, 2001.

[12] J. Z. Fang and G. P. Qiu, “Learning sample subspace with application to face detection,” in *Proc. Int. Conf. Pattern Recog.*, 2004, pp. 423–426.

[13] L. Fei-Fei, R. Fergus, and P. Perona, “A Bayesian approach to unsupervised one-shot learning of object categories,” in *Proc. Int. Conf. Comput. Vis.*, 2003, pp. 1134–1141.

[14] R. Féraud, O. Bernier, J. Viallet, and M. Collobert, “A fast and accurate face detector based on neural networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 1, pp. 42–53, Jan. 2002.

[15] A. Fitzgibbon and A. Zisserman, “Joint manifold distance: A new approach to appearance based clustering,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2003, pp. 26–33.

[16] Y. Freund and R. Schapire, “Experiments with a new boosting algorithm,” in *Proc. 13th Int. Conf. Mach. Learn.*, 1996, pp. 148–156.

[17] Y. Freund and R. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[18] C. Garcia and M. Delakis, “Convolutional face finder: A neural architecture for fast and robust face detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1408–1423, Nov. 2004.

[19] B. Heisele, T. Poggio, and M. Pontil, *Face Detection in Still Gray Images*. Cambridge, MA: MIT, 2000. CBCL Paper 187.

[20] R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, “Face detection in color images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 696–706, May 2002.

[21] D. R. Hundley and M. J. Kirby, “Estimation of topological dimension,” in *Proc. SIAM Int. Conf. Data Mining*, 2003, pp. 194–202.

[22] O. C. Jenkins and M. J Mataric, “Automated derivation of behavior vocabularies for autonomous humanoid motion,” in *Proc. 2nd Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2003, pp. 225–232.

[23] H. L. Jin, Q. S. Liu, and H. Q. Lu, “Face detection using one-class-based support vectors,” in *Proc. 6th IEEE Int. Conf. Autom. Face Gesture Recog.*, 2004, pp. 457–462.

[24] M. H. Law, N. Zhang, and A. K. Jain, “Nonlinear manifold learning for data stream,” in *Proc. SIAM Data Mining*, Orlando, FL, 2004, pp. 33–44.

[25] S. Li and Z. Zhang, “Floatboost learning and statistical face detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1112–1123, Sep. 2004.

[26] C. Liu and H. Y. Shum, “Kullback-Leibler boosting,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2003, pp. 587–594.

[27] C. J. Liu, “A Bayesian discriminating features method for face detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 6, pp. 725–740, Jun. 2003.

[28] L. M. Manevitz and M. Yousef, “One-class SVMs for document classification,” *J. Mach. Learn. Res.*, vol. 2, no. 2, pp. 139–154, 2002.

[29] G. Rätsch, S. Mika, B. Schölkopf, and K.-R. Müller, “Constructing boosting algorithms from SVMs: An application to one-class classification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1184–1199, Sep. 2002.

[30] R. Osadchy, M. Miller, and Y. LeCun, “Synergistic face detection and pose estimation with energy-based model,” in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2005, pp. 1017–1024.

[31] E. Osuna, R. Freund, and F. Girosi, “Training support vector machines: An application to face detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 1997, pp. 130–136.

[32] C. P. Papageorgiou, M. Oren, and T. Poggio, “A general framework for object detection,” in *Proc. 6th Int. Conf. Comput. Vis.*, 1998, pp. 555–562.

[33] K. Pettis, T. Bailey, A. K. Jain, and R. Dubes, “An intrinsic dimensionality estimator from near-neighbor information,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-1, no. 1, pp. 25–37, Jan. 1979.

[34] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[35] S. T. Roweis, L. K. Saul, and G. E. Hinton, "Global coordination of local linear models," in *Advances in Neural Information Processing Systems 14*. Cambridge, MA: MIT Press, 2002, pp. 889–896.

[36] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 23–38, Jan. 1998.

[37] H. A. Rowley, S. Baluja, and T. Kanade, "Rotation invariant neural network-based face detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 1998, pp. 38–44.

[38] H. Schneiderman and T. Kanade, "A statistical method for 3D object detection applied to faces," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2000, pp. 746–751.

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

[40] Y. W. Teh and S. T. Roweis, "Automatic alignment of local representations," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2003, pp. 841–848.

[41] B. J. Tenenbaum, V. Silva, and J. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[42] J. J. Verbeek, N. Vlassis, and B. Krose, "Coordinating principal component analyzers," in *Proc. Int. Conf. Artif. Neural Netw.*, Madrid, Spain, 2002, pp. 914–919.

[43] J. J. Verbeek, N. Vlassis, and B. Krose, "Fast nonlinear dimensionality reduction with topology preserving networks," in *Proc. 10th Eur. Symp. Artif. Neural Netw.*, 2002, pp. 193–198.

[44] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2001, pp. 511–518.

[45] R. Wang, J. Chen, S. Yan, S. Shan, X. Chen, and W. Gao, "Face detection based on the manifold," in *Audio- and Video-Based Biometric Person Authentication*. Berlin, Germany: Springer-Verlag, 2005, pp. 208–218.

[46] R. Xiao, M. J. Li, and H. J. Zhang, "Robust multipose face detection in images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 31–41, Jan. 2004.

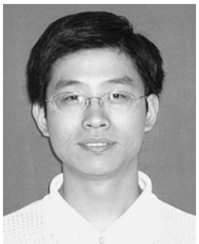
[47] M. H. Yang, "Face recognition using extended Isomap," in *Proc. Int. Conf. Image Process.*, 2002, pp. 117–120.

[48] M. H. Yang, D. Roth, and N. Ahuja, "A SNoW-based face detector," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2000, pp. 855–861.

[49] M. H. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 34–58, Jan. 2002.

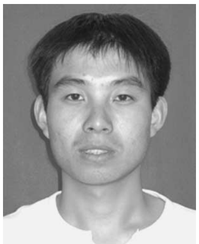
[50] H. Zha and Z. Zhang, "Isometric embedding and continuum Isomap," in *Proc. Int. Conf. Mach. Learn.*, 2003, pp. 864–871.

[51] MIT CBCL Face Database. [Online]. Available: http://cbcl.mit.edu/software-datasets/FaceData2.html



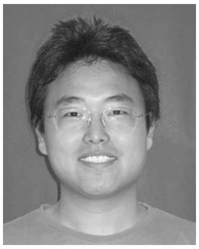
Jie Chen (S'04) received the M.S. degree from the Harbin Institute of Technology, Harbin, China, in 2002. He is currently working toward the Ph.D. degree in the School of Computer Science and Technology, Harbin Institute of Technology.

His research interests include pattern recognition, computer vision, machine learning, and watermarking.



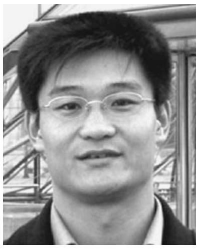
Ruiping Wang received the B.S. degree in applied mathematics from Beijing Jiaotong University, Beijing, China, in 2003. He is currently working toward the Ph.D. degree in the Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing.

He is currently with the Key Laboratory of Intelligent Information Processing, CAS. His research interests include computer vision, pattern recognition, and machine learning.



Shengye Yan (S'06) received the M.S. degree from the Beijing University of Technology, Beijing, China, in 2003. He is currently working toward the Ph.D. degree in computer science in the Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing.

He is currently with the Key Laboratory of Intelligent Information Processing, CAS. His research interests include image processing, pattern recognition, computer vision, and machine learning.



Shiguang Shan (S'01–M'04) received the B.S. and M.S. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1997 and 1999, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing, in 2004.

He is currently an Associate Researcher and serves as the Vice-Director with the Digital Media Center, Institute of Computing Technology, CAS. He is also with the Key Laboratory of Intelligent Information Processing, CAS. He is also the Vice-Director with ICT-ISVision Joint R&D Laboratory for Face Recognition. His research interests cover image analysis, pattern recognition, and computer vision. He is particularly focusing on face-recognition-related research topics.

Xilin Chen (M'00) received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1988, 1991, and 1994, respectively.

He was a Professor with the Harbin Institute of Technology from 1999 to 2005. He was a Visiting Scholar with Carnegie Mellon University, Pittsburgh, PA, from 2001 to 2004. He has been with the Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing, since August 2004. He is also with the Key Laboratory of Intelligent Information Processing, CAS. His research interests are image processing, pattern recognition, computer vision, and multimodal interfaces.

Dr. Chen has served as a program committee member for more than 20 international and national conferences. He has received several awards, including China's State Scientific and Technological Progress Award in 2000, 2003, and 2005 for his research work.

Wen Gao (M'92–SM'05) received the B.Sc. degree from the Harbin University of Science and Technology, Harbin, China, in 1982, the M.Sc. degree from the Harbin Institute of Technology, Harbin, in 1985, all in computer science, and the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991.

He was with the Harbin Institute of Technology, in 1985, where he served as a Lecturer, Professor, and the Head of the Department of Computer Science until 1995. He was with Institute of Computing Technology, Chinese Academy of Sciences (CAS), Beijing, from 1996 to 2005. During his professor career with CAS, he was also appointed as Director with the Institute of Computing Technology, Executive Vice-President with the Graduate School, as well as the Vice President with the University of Science and Technology of China. He is currently a Professor with the School of Electronics Engineering and Computer Science, Peking University, Beijing. He has published four books and over 300 technical articles in refereed journals and proceedings in the areas of multimedia, video compression, face recognition, sign language recognition and synthesis, image retrieval, multimodal interface, and bioinformatics.

Dr. Gao is an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA, an Editor of the *Journal of Visual Communication and Image Representation*, and the Editor-in-Chief of the *Journal of Computer (in Chinese)*. He received the Chinese National Award for Science and Technology Achievement in 2000, 2002, 2003, and 2005.