

Face Detection Based on the Manifold

Ruiping Wang¹, Jie Chen¹, Shengye Yan¹, and Wen Gao^{1,2}

¹ ICT-ISVISION Joint R&D Lab for Face Recognition, Institute of Computing Technology,
Chinese of Academy of Sciences, Beijing, 100080, China
{rpwang, jchen, syyan, wgao}@jdl.ac.cn

² School of Computer Science and Technology,
Harbin Institute of Technology, 150001, China

Abstract. Data collection for both training and testing a classifier is a tedious but essential step towards face detection and recognition. It is a piece of cake to collect more than hundreds of thousands of examples from web and digital camera nowadays. How to train a face detector based on the collected immense face database? This paper presents a manifold-based method to select a training set. That is to say we learn the manifold from the collected enormous face database and then subsample and interweave the training set by the estimated geodesic distance in the low-dimensional manifold embedding. By the resulting training set, we train an AdaBoost-based face detector. The trained detector is tested on the MIT+CMU frontal face test set. The experimental results show that the proposed method based on the manifold is efficient to train a classifier confronted with the huge database.

1 Introduction

Over the past ten years, face detection has been thoroughly studied in computer vision research for its wide potential applications, such as video surveillance, human computer interface, face recognition, and face image database management etc. 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 [31]. Recently, the emphasis has been laid on data-driven learning-based techniques, such as [7, 13, 14, 15, 19, 20, 21, 22, 30]. All of these schemes can be found in the recent survey by Yang [31]. After the survey, one of the important progresses is the boosting-based method proposed by Viola [23] who uses the Haar features for the rapid object detection, and a lot of related works then followed [11, 12, 28].

The performance of these learning-based methods highly depends on the training set, and they suffer from a common problem of data collection for training. It is a piece of cake to collect more than hundreds of thousands of examples from web and digital camera nowadays. How to train a classifier based on the collected immense face database? This paper will give a solution.

In nature, how to train a classifier based on the collected immense face database is a problem of data mining. In this paper we will use the knowledge of the manifold to subsample a small subset from the collected huge face database and then interweave some big hole among the manifold embedding. Manifold can help us to transform the data to a low-dimensional space with little loss of information, which can enable us to visualize data, perform classification and cluster more efficiently. Recently, some representative techniques, including isometric feature mapping (ISOMAP) [25], local

linear embedding (LLE) [17], and Laplacian Eigenmap [1], have been proposed. The ISOMAP algorithm is intuitive, well understood and produces reasonable mapping results [9, 10, 29]. Also, it is supported theoretically [2, 5, 32], which has been developed by [3, 8, 16, 18, 24, 26, 27].

The main contributions of this paper are:

1. Subsample a small but efficient and representative subset from the collected huge face database based on the manifold embedding to train a classifier.
2. Interweave the subsampled manifold embedding to fill in the big holes to complete the training set furthermore.
3. The performance is instable to train a detector based on the random subsampling face set from a huge database. However, a detector trained based on the subsampled face set by the data manifold is not only stable but also can improve the detector performance.

The rest of this paper is organized as follows: After a review of ISOMAP, the proposed subsampling and interweaving method based on the manifold embedding is described in section 2. Experimental results are presented in section 3, followed by discussion in section 4.

2 Subsampling Based on ISOMAP

2.1 ISOMAP Algorithm

The goal of learning the data manifold is to show high-dimensional data in its intrinsic low-dimensional structures and use easily measured local metric information to learn the underlying global geometry of a data set [25]. Given a set of data points $X = \{x_1, \dots, x_n\}$ in a high dimensional space, let $d_X(x_i, x_j)$ be the distance between x_i and x_j ; let $y_i \in R^d$ be the co-ordinates corresponding to x_i and $Y = \{y_1, \dots, y_n\}$. Let $d_Y(y_i, y_j)$ be the distance between y_i and y_j , which is an Euclidean distance in a d -dimensional Euclidean space Y . ISOMAP attempts to recover an isometric mapping from the co-ordinate space to the manifold. The neighborhood is necessary to be specified by ISOMAP. It can be knn -neighborhood, where x_i and x_j are neighbors if $x_i(x_j)$ is one of the k nearest neighbors (knn) of $x_j(x_i)$.

Let the vertex $v_i \in V$ corresponding to x_i ; between v_i and v_j , an edge $e(i, j)$ exists iff x_i is a neighbor of x_j . The weight of $e(i, j)$ is simply $d_X(x_i, x_j)$. And then a weighted undirected neighborhood graph $G = (V, E)$ is constructed. Let $d_G(v_i, v_j)$ denote the length of the shortest path $sp(i, j)$ between v_i and v_j . The shortest paths can be found by the Dijkstra's algorithm, and the shortest paths can be stored efficiently by the predecessor matrix τ_{ij} , where $\tau_{ij} = k$ if v_k is immediately before v_j in $sp(i, j)$. We may call $d_G(v_i, v_j)$ "geodesic distance". That is to say after embedding the high-dimensional data manifold into low-dimensional structures, we can use straight lines in the embedding to approximate the true geodesic path.

2.2 Subsampling Algorithm

As discussed in [25], with the increase of the embedding dimensionality d , the difference between the Euclidean distance in the d -dimensional Euclidean space Y and the true geodesic path decreases. Therefore, after learning its manifold and embedding it in low-dimensionality, we can use the Euclidean distance in the d -dimensional Euclidean space Y to delete some examples from the huge database. And the remained examples can still keep the data's intrinsic geometric structure basically. By this means, we can get a small representative subset of the huge data.

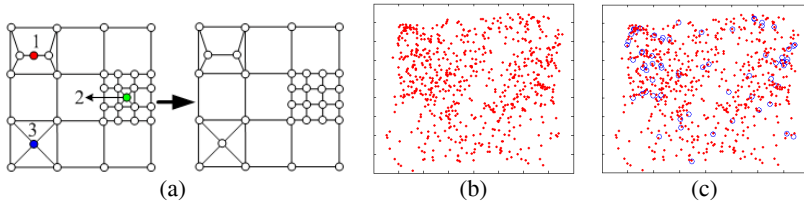


Fig. 1. Subsampling based on the manifold embedding. (a) The schematic of subsampling based on the estimated geodesic distance; (b) the manifold embedding of the 698 raw face images [25]; (c) the results of subsampling based on the estimated geodesic distance

The scheme is demonstrated in Fig. 1 (a). We sort all of the Euclidean distance (i.e., the estimated geodesic distance) between pairs of points y_i and y_j in the d -dimensional Euclidean space Y in increasing order. If the estimated geodesic distance between an example and its neighbor examples is smaller than a given threshold, it will be deleted while its neighbor examples will be reserved. For example, as shown in Fig. 1 (a), the data point 1 and 2 will be deleted when subsampling in the embedding while its neighbors are reserved. As to the data point 3, it is preserved since the estimated geodesic distance between it and its neighbors are bigger than the given threshold. From the figure of top right in Fig. 1 (a), the remained examples can still maintain the data's intrinsic geometric structure basically.

As demonstrated in Fig. 1 (b), it is a two-dimensional projection of 698 raw face images where the three-dimensional embedding of data's intrinsic geometric structure is learned by ISOMAP ($K=6$) [25]. Fig. 1 (c) is the results of subsampling where the data points (blue circle) are deleted and the remained data points are still in red dots.

If we want to subsample 90% examples from a whole set, what we need to do is to delete its 10% examples since their corresponding estimated geodesic distances to their neighbors are in the front of the sorted distance sequence.

2.3 Interweaving Algorithm

To complete the training set furthermore, we fill in the hole among the manifold embedding after the subsampling. The basic idea is as shown in Fig.2. The solid circle points in Fig 2 (a) are the filled examples. How to search these holes in the manifold embedding? In our case, we calculate the *median* of all of the Euclidean distance between pairs of points y_i and its nearest neighbor y_j in the d -dimensional Euclidean space Y . The median is used as the radius of the searching ring as demonstrated in

Fig.2 (b). Moving the searching ring along the embedding, we can get several holes wanted. As shown in Fig 2 (c), the centers of these holes are the places where we generate virtual samples.

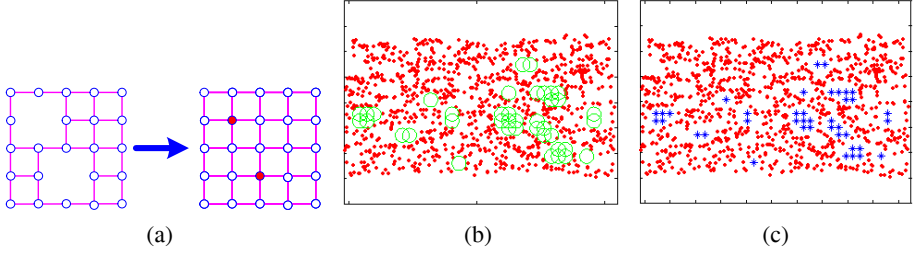


Fig. 2. Interweaving based on the manifold embedding. (a) The schematic of interweaving based on the estimated geodesic distance; (b) the searched holes among the manifold embedding of the Swiss roll [25]; (c) the results of interweaving based on the estimated geodesic distance

Having found the holes in the embedding, the next step is to generate the virtual examples to fill in these holes. In our case, after we have learned the manifold embedding about the face example set, we fill in the holes with some virtual face examples. The basic idea is as following:

1. Perform PCA to the face database, then a coefficient vector \bar{e}_i is computed for each sample f_i in our face database $\{f_1, f_2, \dots, f_n\}$, where n is the number of the face examples;
2. Get the K neighbors $\{f_{n1}, f_{n2}, \dots, f_{nK}\}$ of a virtual example VE_p ($p=1, \dots, m$) and the Euclidean distance $d_Y(VE_p, f_i)(i=1, \dots, K)$ between it and its neighbors in the d -dimensional Euclidean space Y , where m is the number of virtual examples;
3. Calculate the weight $\omega_{pi} = \omega_Y(VE_p, f_i) = 1/d_Y(VE_p, f_i)$, and are normalized:

$$\tilde{\omega}_{pi} = \frac{\omega_{pi}}{\sum_{i=1}^K \omega_{pi}};$$

4. The coefficient vector \bar{e}_p of the virtual example VE_p is generated by the linear combination of the corresponding coefficient vectors $\{\bar{e}_{n1}, \bar{e}_{n2}, \dots, \bar{e}_{nK}\}$:

$$\bar{e}_p = \tilde{\omega}_{p1}\bar{e}_{n1} + \tilde{\omega}_{p2}\bar{e}_{n2} + \dots + \tilde{\omega}_{pK}\bar{e}_{nK};$$
5. Reconstruct the virtual example VE_p with the coefficient vector \bar{e}_p .

Some synthetic virtual samples are shown in table 1, while some synthetic virtual samples and its neighbors are shown in table 2. From the table 1, one can conclude that these synthetic virtual samples look like the real faces very much. From the table 2, one can conclude that the virtual face in the white complexion is constructed by several faces also in the white complexion, while the virtual face in the black is constructed by several faces also in the black and the female by several females.

3 Experiments








































3.1 Detector Based on the MIT Face Database

The data set is consisted of a training set of 6,977 images (2,429 faces and 4,548 non-faces) and the test set is composed of 24,045 images (472 faces and 23,573 non-faces). All of these images are 19×19 grayscale and they are available on the CBCL webpage [33].

Table 1. Original samples vs. synthetic virtual samples

Original samples	
Synthetic virtual samples	

Table 2. Synthetic virtual sample and its corresponding original samples

Synthetic virtual sample:  (in the white complexion)						
Original samples						
weight	0.1172	0.1109	0.0883	0.0816	0.0803	0.0800
Original samples						
weight	0.0780	0.0776	0.0758	0.0734	0.0690	0.0675
Synthetic virtual sample:  (in the black complexion)						
Original samples						
weight	0.1179	0.1132	0.0951	0.0850	0.0838	0.0770
Original samples						
weight	0.0760	0.0716	0.0712	0.0703	0.0692	0.0690
Synthetic virtual sample:  (a female)						
Original samples						
weight	0.0887	0.0874	0.0874	0.0847	0.0843	0.0828
Original samples						
weight	0.0828	0.0825	0.0811	0.0803	0.0796	0.0779

We let $K=6$ when ISOMAP learns the manifold of 2,429 faces in MIT database. The intrinsic dimensionality of the database can be estimated by the inflexion of the curve [25]. As to the MIT face database, its residual variance decreases while the dimensionality d increases as shown in Fig. 3. We can let $d=10$ for the MIT database. However, the residual variance is still 0.097. It is because the face examples in MIT database are too complex, such as different people and variations in poses, facial expressions, lighting conditions.

Note that all of these examples are performed by histogram equalization before learning the manifold. It is because all examples to train a classifier are needed histogram equalization which maps the intensity values to expand the range of intensities.

In order to study the relationship between the distribution of the training set and the trained detector, we subsample the MIT face database by 90%, 80% and 70% (named as ISO90, ISO80, ISO70 later) as discussed in section 2.2. Subsampling by 90% is to

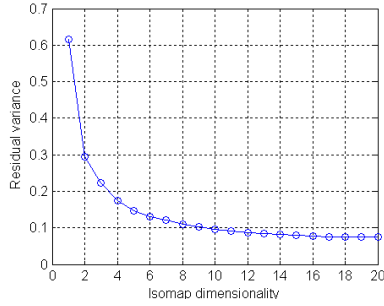


Fig. 3. The residual variance of ISOMAP on the MIT face database

say we reserve 90% examples of the database and the same meaning of 80% and 70%. Note that ISO70 is a subset of ISO80 and ISO80 is a subset of ISO90 in fact.

The three subsampled face sets together with the non-face are used to train three classifiers based on the AdaBoost as demonstrated in [23]. And then they are tested on the test set of MIT database. The ROC curves of these three classifiers are shown in Fig. 4 (a). From these ROC curves, one can conclude that the detector trained by ISO90 is the best of all and improves the performance of the detector distinctly compared with the detector even by the entire face examples in MIT database. Even the detector trained on ISO70 is a little better than the detector trained on the entire examples. Some possible reasons: the first one is the examples of ISO90 distribute evenly in the example space and has no example congregate compared with the whole set; the second is that the outliers in the whole set deteriorate its performance which has been discarded during the manifold learning [25] (During the ISOMAP learning, we get 30 outliers.).

However, random subsampling from the MIT database is not so lucky. We choose four subsets randomly-sampled from the MIT database and each subset has the same number of examples as in ISO90. After trained on these four sets, they are also tested on the same test set. The ROC curves are shown in Fig. 4 (b). In this figure, we plot the resulting ROC curves of detectors on the whole set, ISO90, and two randomly chosen subsets. Herein, the curve “90.6% examples based on the random subsampling $n1$ ” and the curve “90.6% examples based on the random subsampling $n2$ ” represent the best and the worst results of these four random sampling cases. From these ROC curves, one can conclude that the detector based on ISO90 is still the best of all and the results based on random subsampling is much instable. We also think that the evenly-distributed examples and no outliers contribute to this kind of results.

After the subsampling, we interweave the manifold embedding as discussed in section 2.3. As shown in Fig. 5 (a), we add the different number of virtual examples in the set ISO90. There are 100 or 500 examples are added into ISO90, respectively. One can conclude that a few numbers of added examples is valuable for training a detector. When the number is up to 500, it will be deteriorate. As shown in Fig. 5 (b), we change the radius of the searching ring. The first 100 examples are searched by the radius equal to the *median*, while the second 100 examples are searched by the radius equal to the $1.1 \times \text{median}$ and the third 100 equal to the $1.2 \times \text{median}$. One can conclude when the radius of the searching ring is equal to the *median*, the added 100 examples is most valuable for training a detector.

3.2 Detector Based on the Huge Database

To compare the performance difference on different training sets further, we also use another three different face training sets. The face-image database consists of 10,000 faces (collected from web, video and digital camera), which cover wide variations in poses, facial expressions and also in lighting conditions. To make the detection method robust to affine transform, the images are often rotated, translated and scaled [6]. After these preprocessing, we get 90,000 face images which constitute the whole set. The first group is composed of 14,000 face images which are sampled by the ISOMAP (called ISO14000, here). The second and third group are also composed of 14,000 face images which are random subsampling examples from the whole set (named Rand1-14000 and Rand2-14000, respectively).

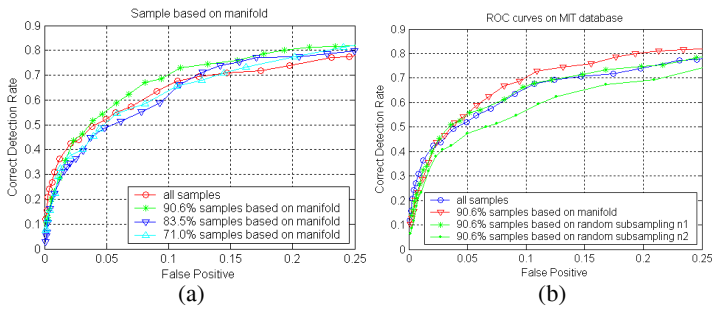


Fig. 4. The ROC curves on the MIT test set. (a) Using the subsampling face example sets based on the manifold embedding and the whole set as training set for a fixed classifier. (b) Using the subsampling face example sets based on the manifold embedding, two random sampling sets and the whole set as training set for a fixed classifier

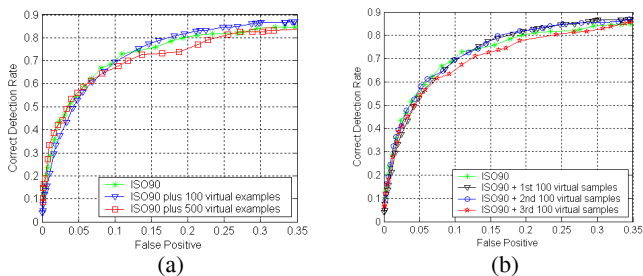


Fig. 5. The ROC curves on the MIT test set using the interweaving face example sets based on the manifold embedding. (a) Add the different number of virtual examples in the training set ISO90. (b) Change the radius of the searching ring

It is hard to learn the manifold from 90,000 examples by the ISOMAP because it needs to calculate the eigenvalues and eigenvectors of a $90,000 \times 90,000$ matrix. In order to avoid this problem, as demonstrated in Fig. 6, we divide the whole set into 30 subsets and each subset has 3,000 examples. We get 1,000 examples by the proposed method from each subset and then incorporate every three subsampled sets into one subset. We will have 10 subsets with the total 30,000 examples. With the same proce-

ture, we can get 1,400 examples by the proposed method from each incorporated subset and then incorporate all subsampled examples into one set with the total 14,000 examples.

The non-face class is initially represented by 14,000 non-face images. Each single classifier is then trained using a bootstrap approach similar to that described in [22] to increase the number of negative examples in the non-face set. The bootstrap is carried out several times on a set of 13,272 images containing no faces.

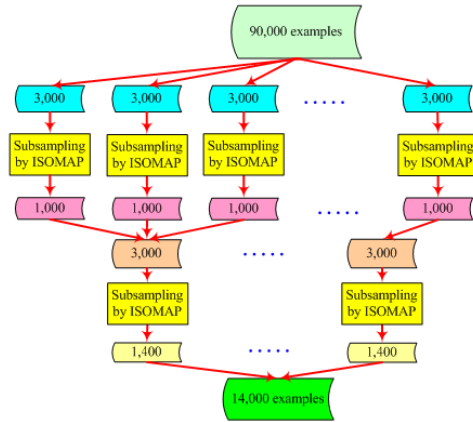


Fig. 6. Subsampling procedure by ISOMAP to get 14,000 examples from 90,000 examples

The resulting detectors, trained on the three different sets, are evaluated on the MIT+CMU frontal face test set which consists of 130 images showing 507 upright faces [19]. The detection performances on this set are compared in Fig. 7 (a). From these ROC curves one can conclude that the detector based on ISO14000 is the best of all and the results based on random subsampling is also much instable. During the ISOMAP learning, we get 838 outliers. We think that the evenly-distributed examples and no outliers contribute this kind of results, again.

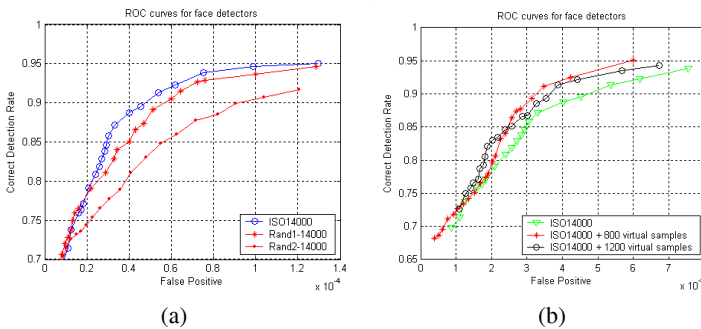


Fig. 7. The ROC curves for the trained detectors. (a) Train detector based on the sampled training set by the ISOMAP and the random subsampling training set. (b) Train detector by adding the virtual samples

Based on the subsampled training set ISO14000, we add some virtual examples (800, 1200 respectively) by the proposed method. As shown in Fig. 7 (b), the detectors by ISO14000 together with adding virtual example outperform the detector only by ISO14000. Some results of our trained detector based on ISO14000 + 800 examples are shown in Fig.8.



Fig. 8. Some results of our trained detector

4 Conclusion

In this paper, we present a novel manifold-based method to subsample a small but efficient and representative training subset from the collected enormous face database. After learning the manifold from the collected face database, we subsample the training set by the estimated geodesic distance in the manifold embedding and then interweave the big holes in the embedding. An AdaBoost-based face detector is trained on the resulting training set in the low-dimensional manifold embedding, and then we test it on the MIT+CMU frontal face test set. Compared with the AdaBoost-based face detector using random subsampling examples, the detector trained by the proposed method is more stable and achieve better face detection performance. We conclude that the evenly-distributed examples, due to the subsampling training set based on the manifold embedding, and no outliers, discarded during the manifold learning, contribute to the improved performance. The added virtual examples can improve the performance of the detector further.

Acknowledgement

This research is partially sponsored by Natural Science Foundation of China under contract No. 60473043, National Hi-Tech Program of China (No. 2003AA142140), and ISVISION Technologies Co., Ltd.

References

1. M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Inform. Proc. Systems 14*, pp.585-591. MIT Press, 2002.

2. M. Bernstein, V. de Silva, J. Langford, and J. Tenenbaum. Graph approximations to geodesics on embedded manifolds. *Technical report, Stanford University*, 2000.
3. M. Brand. Charting a manifold. In *Advances in Neural Information Proc. Systems 15*, pp. 961-968. MIT Press, 2003.
4. J. Chen, X. Chen and W. Gao. Expand Training Set for Face Detection by GA Resampling. *The 6th IEEE Intern. Conf. FG2004*. pp. 73-79. 2004.
5. D. L. Donoho and C. Grimes. When does ISOMAP recover natural parameterization of families of articulated images? *Technical Report 2002-27., Stanford University*, 2002.
6. B. Heisele, T. Poggio, and M. Pontil. Face Detection in Still Gray Images. *CBCL Paper #187*. MIT, Cambridge, MA, 2000.
7. R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain, "Face detection in color images," *IEEE Trans. Pattern Anal. Machine Intell.*, pp.696-706, 2002.
8. D. R. Hundley and M. J. Kirby. Estimation of topological dimension. In *Proc. SIAM International Conference on Data Mining*, 2003.
http://www.siam.org/meetings/sdm03/proceedings/sdm03_18.pdf
9. O. C. Jenkins and M. J. Mataric. Automated derivation of behavior vocabularies for autonomous humanoid motion. In *Proc. of the Second Int'l Joint Conference on Autonomous Agents and Multiagent Systems*, Melbourne, Australia, July 2003.
10. M. H. Law, N. Zhang, A. K. Jain. Nonlinear Manifold Learning for Data Stream. In *Proc. of SIAM Data Mining*, pp. 33-44, Florida, 2004.
11. S. Z. Li, L. Zhu, Z.Q. Zhang, A. Blake, H. J. Zhang, and H. Shum. Statistical Learning of Multi-View Face Detection. In *Proc. of the 7th ECCV*. 2002.
12. C. Liu, H. Y. Shum. Kullback-Leibler Boosting. *Proceedings of the 2003 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'03)*. 2003.
13. C. J. Liu. A Bayesian Discriminating Features Method for Face Detection, *IEEE Trans. Pattern Anal. and Machine Intel.*, pp. 725-740. 2003.
14. E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proc. IEEE Conf. on CVPR*, pp. 130-136. 1997,
15. C. P. Papageorgiou, M. Oren, and T. Poggio, "A general framework for object detection," in *Proc. 6th Int. Conf. Computer Vision*, pp.555-562. 1998,
16. K. Pettis, T. Bailey, A. K. Jain, and R. Dubes. An intrinsic dimensionality estimator from near-neighbor information. *IEEE Trans. of Patt. Anal. and Machine Intel.*, pp.25-36, 1979.
17. S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290: pp.2323-2326, 2000.
18. S. T. Roweis, L. K. Saul, and G. E. Hinton. Global coordination of local linear models. In *Advances in Neural Information Processing Systems 14*, pp. 889-896. MIT Press, 2002.
19. H. A. Rowley, S. Baluja, and T. Kanade. Neural Network-Based Face Detection. *IEEE Tr. Pattern Analysis and Machine Intel.* pp. 23-38. 1998.
20. H. A. Rowley, S. Baluja, and T. Kanade. Rotation Invariant Neural Network-Based Face Detection. *Conf. Computer Vision and Pattern Rec.*, pp. 38-44. 1998.
21. H. Schneiderman and T. Kanade. A Statistical Method for 3D Object Detection Applied to Faces. *Comp. Vision and Pattern Recog.*, pp. 746-751. 2000.
22. K. K. Sung, and T. Poggio. Example-Based Learning for View-Based Human Face Detection. *IEEE Trans. on PAM*. pp. 39-51. 1998.
23. P. Viola and M. Jones. Rapid Object Detection Using a Boosted Cascade of Simple Features. *Conf. Comp. Vision and Pattern Recog.*, pp. 511-518. 2001.
24. Y. W. Teh and S. T. Roweis. Automatic alignment of local representations. In *Advances in Neural Information Processing Systems 15*, pp. 841-848. MIT Press, 2003.
25. B. J. Tenenbaum, V. Silva, and J. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, Volume 290, pp.2319-2323, 2000
26. J.J. Verbeek, N. Vlassis, and B. Krose. Coordinating principal component analyzers. In *Proc. of International Conf. on Artificial Neural Networks*, pp. 914-919, Spain, 2002.

27. J.J. Verbeek, N. Vlassis, and B. Krose. Fast nonlinear dimensionality reduction with topology preserving networks. In *Proc. 10th European Symposium on Artificial Neural Networks*, pp.193-198, 2002.
28. R. Xiao, M. J. Li, H. J. Zhang. Robust Multipose Face Detection in Images, *IEEE Trans on Circuits and Systems for Video Technology*, Vol.14, No.1 pp. 31-41. 2004,
29. M.-H. Yang. Face recognition using extended ISOMAP. In *ICIP*, pp.117-120, 2002.
30. M. H. Yang, D. Roth, and N. Ahuja. A SNoW-Based Face Detector. *Advances in Neural Information Processing Systems 12*, MIT Press, pp. 855-861. 2000.
31. M. H. Yang, D. Kriegman, and N. Ahuja. Detecting Faces in Images: A Survey. *IEEE Tr. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 34-58. 2002.
32. H. Zha and Z. Zhang. Isometric embedding and continuum ISOMAP. In *ICML*, 2003. <http://www.hpl.hp.com/conferences/icml2003/papers/8.pdf>
33. <http://www.ai.mit.edu/projects/cbcl/software-dataset/index.html>