Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

# A Semi-Supervised Support Vector Machine Based Algorithm for Face Recognition

Wei-Shan Yang†, Chun-Wei Tsai*, Keng-Mao Cho†, Chu-Sing Yang‡, Shou-Jen Lin† and Ming-Chao Chiang*
†Department of Computer and Communication Engineering,
National Cheng Kung University, Tainan, Taiwan, R.O.C.
*Department of Computer Science and Engineering,
National Sun Yat-sen University, Kaohsiung, Taiwan, R.O.C.
‡Department of Electrical Engineering,
National Cheng Kung University, Tainan, Taiwan, R.O.C.
Email: q3696424@mail.ncku.edu.tw, cwtsai87@gmail.com, q3897130@mail.ncku.edu.tw,
csyang@ee.ncku.edu.tw, mcchiang@cse.nsysu.edu.tw

***Abstract*—Most, if not all, of the researches in support vector machine (SVM) based face recognition algorithms have generally presumed that the classifier is static and thus unscalable, due to the fact that SVM is a supervised learning method. This paper introduces a novel SVM based face recognition method—by dynamically adding "new" faces of existing or new persons into the face database—which circumvents these difficulties. In other words, the proposed algorithm is able to learn and recognize faces that are not in the face database before. The paper presents the theory and the experimental results using the new approach. Our experimental results indicate that the accuracy rate of the proposed algorithm ranges from 91% up to 100% and outperforms all the others.**

***Index Terms*—support vector machine, face recognition.**

## I. Introduction

A face recognition system is generally composed of three modules: face detection, feature extraction, and face recognition [1]. The face detection module is responsible for segmentation of faces, the feature extraction module for feature extraction from the face regions, and the face recognition module for recognition and verification.

Most of the recent researches on face recognition are focused on how to improve the accuracy rate of a face recognition system. These researches basically fall into two groups. One group is focused on developing a more effective feature extraction module to reduce the influence of illumination, position, orientation, scale, and expression [2], [3], [4], [5], [6], [7], [8]. The other group is focused on using a more effective classification method to enhance the quality of the end results of a face recognition system. The algorithms proposed include Bayesian Framework [9], [10], Nearest Neighbor (NN) [11], [12], [13], Genetic Algorithm (GA) [14], [15], Hidden Markov Models (HMM) [16], [17], and Support Vector Machine (SVM) [12], [18], [19], [20], [21]. Nowadays, Support Vector Machine (SVM) has been widely used in solving the classification problem for face recognition for a very simple reason. SVM can be used to partition the data that are *non-linearly separable* such as human face images. Another reason is that SVM provides a very high accuracy rate for the human face classification problem, especially when the data set is complicated.

However, one of the major problems with the SVM based face recognition system is that the classifier,[1] once created, is not allowed to be changed—be it for recognition or verification. The consequence is that all the facial data have to be available at the training stage insofar as a traditional SVM is concerned. Unfortunately, more often than not, not all the facial data can be collected at once in practice.

The proposed algorithm resembles the semi-supervised learning (SSL) [22], [23], which assumes that the training data set is composed of both labeled and unlabeled data. The labeled data are used to create the initial classifier as we did in the paper while the unlabeled data are used to enhance the "performance" of the classifier but the number of classes in the classifier will remain intact. The proposed algorithm can eventually increase not only the number of images in the face database but also the number of classifiers in the classifier, all on the fly. Similar to SSL, the proposed algorithm faces some of the problems SSL faced, namely, the unlabeled data may degrade the performance of the classifier.

The remainder of the paper is organized as follows. Section II briefly discusses SVM and how it can be used for face recognition. It also gives the problem definition. Section III presents the proposed algorithm and explains how the proposed algorithm is designed. The experimental results are given in Section IV. We conclude our work and give future directions in Section V.

## II. Related Work and Problem Definition

### A. Support Vector Machine for Face Recognition

Support Vector Machine (SVM) is a relatively recent learning technique based on the structural risk minimization principle [24]. The basic idea of SVM [25], [26], [27] is to construct a hyperplane with the largest margin to classify a set

[1]As far as this paper is concerned, unless stated otherwise, by "classifier", we mean either a tree of classifiers or a set of classifiers or even a standalone classifier.



978-1-4244-2794-9/09/$25.00 ©2009 IEEE

of data drawn from a known or unknown distribution into two classes. Or more precisely and mathematically, the basic idea of SVM is to assign a label $y_i \in \{-1, +1\}$ to each pattern $x_i$ in the training data set. For the training patterns that are linearly separable, the optimal separating hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$ where $\mathbf{w}$ is the normal to the hyperplane and $b$ a scalar constant that can be found by minimizing

$$\frac{\|\mathbf{w}\|^2}{2} \quad \text{subject to } y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0$$

for all $i = 1, 2, \ldots, n$. For the training data that are non-linearly separable, slack variables $\xi_i$, penalty parameter $C$, and kernel function $\Phi$ can be introduced, and the optimal separating hyperplane can be found by minimizing

$$\frac{\|\mathbf{w}\|^2}{2} + C\sum_{i=1}^{n}\xi_i \quad \text{subject to } y_i(\Phi(\mathbf{x}_i) \cdot \mathbf{w} + b) - 1 + \xi_i \geq 0$$

for all $i = 1, 2, \ldots, n$.

Although relatively new, SVM based classification algorithms have been widely used in solving the face recognition problem [12], [18], [19]. Basically, SVM is a binary classification algorithm. When used in solving the multi-class problem [26], the structures of one-against-all, one-against-one, binary decision, directed acyclic graph, and so on can be used. In [28], Benabdeslem presents a Dendogram-based Support Vector Machine (DSVM) for constructing the classifier. DSVM first computes the mean of each group and then uses all the means to perform the hierarchical clustering. DSVM can efficiently reduce the number of classifiers from $\binom{n}{2}$ down to $n-1$, as Fig. 1 shows. It will in turn reduce the time classification takes from $O(n)$ or even $O(n^2)$ down to $O(\log n)$.

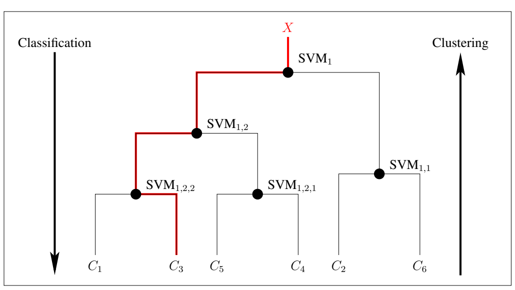



Fig. 1. Example showing how a DSVM classifier is used to recognize a test pattern $X$.

In [12], Kang presents a Support Vector Domain Description (SVDD) method for multi-class classification. By avoiding the repeated use of data, SVDD can also reduce the training time. In [18], Mu points out that the performance of SVM is sensitive to the setting of kernel and regularization parameters. They assumed that the SVM parameter tuning problem can be viewed as a weighted least-square minimization problem. Then, the extended Kalman filter (EKF) algorithm can be applied to solve the multiple parameter tuning 2-norm Support Vector Machine ($L_2$-SVM) problem. In [19], Hong presents an emotion recognition system that can adapt to new facial data. They collect facial expression samples to train the SVM classifier. Only if new facial expressions are recognized erroneously, will the erroneously recognized data and the data that are nearest to the original hyperplane be used to retrain the classifier.

### *B. The Problem of Newly Created Classes*

Apparently, training the classifier [29], [30], [31], [32] has become more and more important for the face recognition problem, especially when either SVM or SVM based algorithm is used and the face database is either large or non-static. This can be easily justified by the observation that as the training data set becomes dynamic, it is only a subset of the training data set that will be available for creating the initial classifier. In other words, not all the training data need to be involved in the creation of the classifier at the very beginning. This would save both the memory space and the computation time, by adding new classes into the classifier thus providing an approach to solving the problem of not having all the facial data during the initial training stage.

In [33], $S^3$VM is presented with branch and bound technique. When new training data arrive, only the neighborhood of these new data will be considered for updating the $S^3$VM. Luo [34] uses fixed-partition method to partition the data set into fixed size subsets and then train them one by one. Luo further uses error-driven method to decide whether or not to retrain the data. The growth of the memory requirements for SVM is controlled by excluding those which can be expressed as a linear combination of the others in the feature space. Cauwenberghs [35] constructs the solution recursively, one point at a time. The correctness of the solution is guaranteed by retaining the Kuhn-Tucker conditions when a new data point is added to the solution.

However, if the system only accepted that the addition of new patterns that should be in the group of the training data, it will not be satisfied the conditions of the real world system. Fig. 2 gives an example of a classifier, which is composed of two classes of data. Fig. 3 shows how the classifier is trained. Upon receiving of new data, the system will first add the new data to the face database and then train the classifier—by adjusting the setting of the hyperplanes. Fig. 3(a) shows an example that requires no new classes to be created because the new training data belong to existing classes. Figs. 3(b) to (d) show other examples that require a new classifier to be added. Note that only some of the possible cases that the proposed algorithm may face upon receiving of new data are shown in Fig. 3. Together, the examples described in Figs. 3(a)–(d) show how the system can use a subset of training data to create an initial classifier and then add new classes to the classifier. In summary, it is the combination of these two methods that provide the SVM a more efficient way to handle a face database that is either large or non-static.

The problem is defined as follows:

**Input:** First, a set of labeled data $D_L$. Then, a stream of unlabeled $D_U$ entering the system one after another.

**Output:** A classifier for all the training data $D = D_L \cup D_U$.

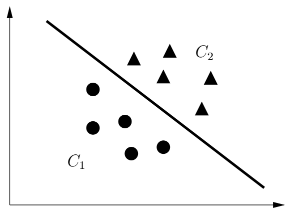



Fig. 2. Example showing two classes of training data separated by a hyperplane.

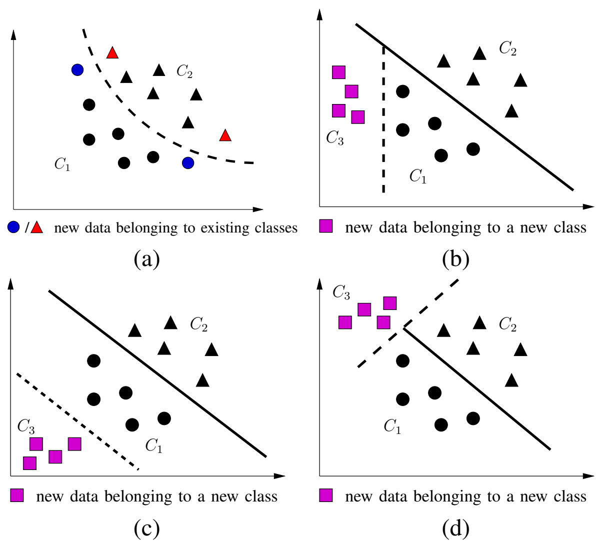



Fig. 3. An example illustrating how the classifier is updated. (a) new training data belong to existing classes. (b)–(d) new training data belong to a new class.

Note that data in $D_L$ are labeled; the training can be considered as supervised. On the other hand, data in $D_U$ are unlabeled; the training can be considered as unsupervised. Thus, the name "Semi-SVM (Semi-Supervised Support Vector Machine)," or Semi-SVM for short.

## III. The Proposed Algorithm

### A. Semi-Supervised SVM

Fig. 4 gives an outline of the Semi-SVM algorithm that is basically composed of two modules: (1) supervised learning and (2) unsupervised learning. The supervised learning module takes the responsibility of creating the initial classifier as well as a set of half-hyperplanes the purpose of which is to help the unsupervised learning module identify if an unlabeled datum is eventually a member of an existing or a new class. The unsupervised learning module plays the role of training the classifier, by either updating existing classes or creating new classes if necessary.

The supervised learning module starts on line 2. First, the candidate pool $P$ and the classifier $T$ are each initialized to be an empty set on line 2. Then, the initial classifier $T$ for the set of labeled data $D_L$ is constructed on line 3 using DSVM [28] for DSVM can save a great deal of the training time as we discussed earlier. After that, the proposed algorithm uses the one-against-all SVMs to partition $D_L$ into a set of $q$ half-hyperplanes $R = \{R_1, R_2, \ldots, R_q\}$ each of which is associated with a classifier in $T$ on line 4, where $q$ is the number of labeled classes. That is, $T = \{T_1, T_2, \ldots, T_q\}$ and $R_i$ corresponds to $C_i$ for all $i$. As stated earlier, the sole purpose of the set of half-hyperplanes is to help Semi-SVM identify if a new pattern should be assigned to either an existing or a new class.

The unsupervised learning module starts on line 6. First, the number of new classes $\kappa$ is reset to zero on line 6. Then, for each of the unlabeled datum $d_j$, it will be identified by Semi-SVM as either belonging to an existing class or to a new class on line 8. First, the distances between $d_j$ and all the classes currently in the classifier $T$ are computed. Then, the class $C_i$ that is closest to $d_j$ is chosen as the class of $d_j$. After that, $d_j$ is checked to see if it is also in the half-hyperplane $R_i$. If not, it will be assigned to the candidate pool $P$ on line 9. On line 10, if a brand new class is needed to be created,[2] the number of new classes $\kappa$ will be incremented. Otherwise, it will be assumed to be a member of the class $C_i$ and will be added into the classifier $T$ on line 12. This loop (lines 7–13) will be repeated until no data are available. After that, the patterns in the candidate pool $P$ will be partitioned into $\kappa$ classes on line 14. Then, each of the new classes with more than three patterns in it is merged to the classifier $T$. Classes with no more than three patterns are assumed to be either noise or outlier and are put back to the candidate pool $P$—to be considered again along with new patterns put in the pool in the next round of the training process—on line 16. Then, the classifier $T$ is output on line 17. Finally, the system will be waiting for new data to arrive. and upon receiving of new data, the system will get back to 6 for another round of the training process.

### B. An Example

In this section, we present a simple example to illustrate exactly how Semi-SVM works. As described in Fig. 5, the classifier is initially composed of three classes and then grown to five classes at the end. The details of how it is constructed are given below.

First, the supervised learning module of Semi-SVM as shown in Fig. 4 is applied to construct an $n - 1 = 3 - 1 = 2$ node classifier as (i) of Fig. 5(a) shows. Then, as (ii) of Fig. 5(a) indicates, one-against-all SVMs is performed to partition the training patterns into a set of half-hyperplanes $R = \{R_1, R_2, R_3\}$ each of which are associated with the corresponding class shown in (i) of Fig. 5(a). That is, $R_i$ is associated with $C_i$ for all $i = 1, 2, 3$. As noted earlier, the sole purpose of $R$ is to help Semi-SVM identify if an unlabeled datum is a member of an existing or a new class.

Then, the unsupervised learning step of Semi-SVM as shown in Fig. 5 is applied to identify an unlabeled datum (or pattern) to see if it is a member of an existing or a new class.

[2] That is, if $d_j$ and $d_{j+1}$ belong to the same new class, then only one new class will be created, and the number of new classes $\kappa$ will remain the same.

```
1  // 1. supervised learning
2  Let P = φ and T = φ
3  Create the initial classifier T for the set of labeled data D_L
4  Partition D_L into a set of half-hyperplanes R using the one-against-all SVMs classification
5  // 2. unsupervised learning
6  Let κ = 0.
7  For each unlabeled datum d_j ∈ D_U
8      If d_j is most similar to C_i but not also in the associated half-hyperplane R_i
9          Add d_j to the candidate pool P
10         If a brand new class is required for d_j, then κ = κ + 1
11     Else
12         Incrementally add d_j into the classifier T
13 End
14 Partition the patterns in P to a set of κ disjoint classes
15 Merge each of the new classes with more than three patterns to T
16 Put each of the new clusters with no more than three patterns back to P
17 Output the classifier T
18 Wait for new data. Upon receiving of new data, go to line 6
```

Fig. 4. Outline of the Semi-SVM algorithm.

The most important issue as far as the unsupervised learning module is concerned is how to deal with these two situations. In the paper, the DSVM classifier plays the role of finding the class that is most similar to the newly arrived pattern as shown in (i) of Fig. 5(b). Then, the set of regions $R$ resulting from the one-against-all SVMs is used to determine if the newly arrived pattern is a member of a class currently in the classifier $T$, as (ii) of Fig. 5(b) shows, or it is supposed to be in a new class, as (iii) of Fig. 5(b) indicates. More precisely, if the newly arrived pattern is in the half-hyperplane $R_i$ associated with the class $C_i$, then the pattern is considered as a member of $C_i$. For example, if a pattern is classified as a member of class $C_1$ by the DSVM classifier, but it is not also identified as in the half-hyperplane $R_1$ associated with $C_1$. The proposed algorithm will consider the pattern as belonging to a new class. The situation would look like those given in Fig. 3(b) to (d). If Semi-SVM identifies the pattern as belonging to a new class, the proposed algorithm will put it into the candidate pool $P$ as shown in (iii) of Fig. 5(b). Otherwise, the pattern will be added to the classifier as shown in (ii) of Fig. 5(b). Finally, a partitional clustering algorithm such as $k$-means is employed to partition the patterns in $P$ into $\kappa$ clusters as shown in (iv) of Fig. 5(b). Then, classes with no more than three patterns will be considered as either noise or outlier and will be put back to the candidate pool $P$. Each of the remaining clusters will be merged to the node that is closest to it, as (v) of Fig. 5(b) shows.

## IV. Experimental Results

In the paper, we evaluate one-against-one SVM, DSVM [28], and Semi-SVM. All empirical analyses are run on a Pentium-D machine with 768 MB of memory. The operation system is Fedora Core 8, and the programs are written in C and compiled by gcc version 4.3.0. The details of the data sets and the evaluation of these algorithms will be given in the following sections. The kernel function $\Phi$ used in this paper is the radial basis function (RBF) kernel.

### A. Data Sets and Parameter Settings

For the purpose of evaluating the performance of the proposed algorithm, the ORL face database [36] is used. It contains 400 face images from 40 distinct subjects with 10 frontal exposures of each assuming different facial expressions, lighting, and slight orientation changes. In the paper, we test three cases each of which consist, respectively, of 10, 20, and 40 classes. The 10-class case contains 8 classes of labeled data and 2 classes of unlabeled data; the 20-class case 16 classes of labeled data and 4 classes of unlabeled data; and the 40-class case 32 classes of labeled data and 8 classes of unlabeled data. For all the three cases tested, 70% of each class of the labeled data are used to construct the initial classifier. For instance, for the 10-class case, $8 \times 7 = 56$ labeled images are used. In addition to classes of unlabeled data, the remaining 30% of each class of the labeled data are used as the unlabeled data for the purpose of unsupervised training.

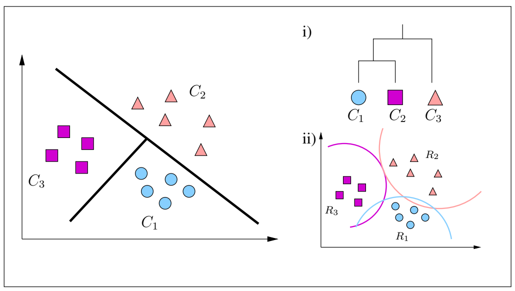
### B. Simulation Results

Table I compares the performance of Semi-SVM with one-against-one SVM and DSVM. Table I shows that the quality of the end results of DSVM is very close to that of one-against-one SVM, but DSVM requires only $n-1$ SVM classifiers instead of $\binom{n}{2}$. The end results of Semi-SVM are exactly the same as those of DSVM for all the three cases we tested.

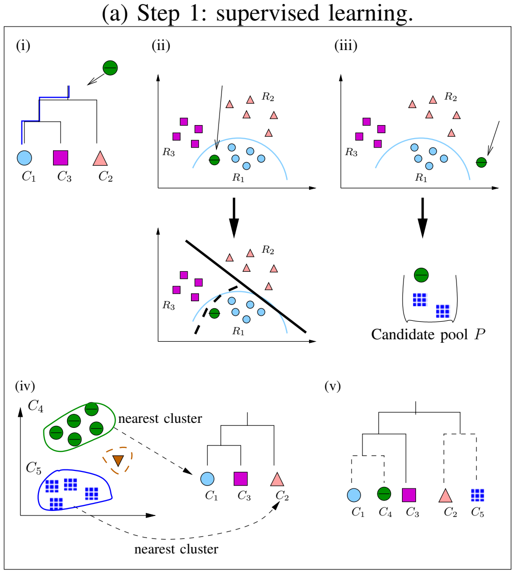
Since only 70% of the labeled data are used for the purpose of supervised training, not all the label data are correctly recognized. For instance, (74 labeled, 19 unlabeled) in the last column of Table I indicates that for the 10-class case, only 74 out of 80 (i.e., 92.5%) labeled images are correctly

TABLE I
SIMULATION RESULTS OF DSVM AND SEMI-SVM.

| classes | one-against-one | | DSVM | | Semi-SVM | |
|---|---|---|---|---|---|---|
| | images | accuracy | images | accuracy | images | accuracy |
| 8 (all labeled) | 80 / 80 | 100.00% | 80 / 80 | 100.00% | 80 / 80 | 100.00% |
| 10 (8 labeled + 2 unlabeled) | — | — | — | — | 93 / 100<br>( 74 labeled, 19 unlabeled) | 93.00%<br>(92.50% labeled, 95.0% unlabeled) |
| 16 (all labeled) | 157 / 160 | 98.13% | 157 / 160 | 98.13% | 157 / 160 | 98.13% |
| 20 (16 labeled + 4 unlabeled) | — | — | — | — | 188 / 200<br>(150 labeled, 38 unlabeled) | 94.00%<br>(93.75% labeled, 95.0% unlabeled) |
| 32 (all labeled) | 312 / 320 | 97.50% | 312 / 320 | 97.50% | 312 / 320 | 97.50% |
| 40 (32 labeled + 8 unlabeled) | — | — | — | — | 364 / 400<br>(294 labeled, 70 unlabeled) | 91.00%<br>(91.86% labeled, 87.5% unlabeled) |




(a) Step 1: supervised learning.




(b) Step 2: unsupervised learning.

Fig. 5. A simple example illustrating how Semi-SVM works. See the text for more detailed explanation.

recognized, and only 19 out of 20 (i.e., 95%) unlabeled images are correctly recognized.

The simulation results show that the accuracy rate of Semi-SVM is no less than 91% for all the cases we tested. The results of one-against-one SVM and DSVM are not provided because they do not handle unlabeled data. For example, if the one-against-one SVM classifier is constructed using 8 classes of labeled data, but the test data consist of 10 classes two of which are unlabeled. Then, 20% of the test data will be identified by the one-against-one SVM classifier as members of the eight classes. The accuracy rate of SVM and DSVM will be less than 80%. For Semi-SVM, the accuracy rate is about 91%, at least 11% higher than SVM and DSVM. On the other hand, these results showed that as the number of unlabeled classes increases, the accuracy rate of Semi-SVM will decrease much slower than that of the traditional SVM.

## V. CONCLUSION

In practice, it is almost impossible to collect all the facial data at once. Even if it is possible to collect all the facial data, it is often extremely expensive to label them because a great deal of efforts are required. As such, a face recognition system that is capable of semi-supervised learning would significantly reduce the cost of such a system. We propose a dynamic semi-supervised support vector machine based algorithm, called Semi-SVM, to deal with the classification problem the input of which consist of both labeled and unlabeled data. It is fundamentally different from traditional SVM in that the face database need not hold all the facial data of each person at the very beginning. As such, Semi-SVM provides a flexible solution to the real-world problem. Our simulation results showed that the accuracy rate of the proposed algorithm is very close to that of the traditional SVM based classification algorithms, which require that the person to be recognized be in the face database. Moreover, the user of the proposed face recognition system is now able to label new faces in a group basis instead of one by one. As for the performance of Semi-SVM, if no new class is ever needed to be created, the accuracy rate of the proposed algorithm is exactly the same as DSVM. Otherwise, both SVM and DSVM are not able to solve the problem, but Semi-SVM can solve such a problem.

## ACKNOWLEDGMENT

This work was supported by National Science Council, R.O.C., under contact number 97-2218-E-006-015.

## REFERENCES


[1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, 2003.

[2] O. Arandjelovic and A. Zisserman, "Automatic face recognition for film character retrieval in feature-length films," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 30, 2005, pp. 860–867.

[3] G. J. Edwards, C. J. Taylor, and T. F. Cootes, "Improving identification performance by integrating evidence from sequences," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 1999, pp. 486–491.

[4] K. W. Bowyer, K. Chang, and P. Flynn, "A survey of approaches and challenges in 3d and multi-modal 3d + 2d face recognition," *Elsevier Computer Vision and Image Understanding*, vol. 101, no. 1, pp. 1–15, 2006.

[5] A. M. Bronstein, M. M. Bronstein, and R. Kimmel, "Expression-invariant 3d face recognition," in *Proceedings of the Audio and Video-based Biometric Person Authentication*, 2003, pp. 62–69.

[6] S. Gundimada and V. Asari, "Adaptive confidence level assignment to segmented human face regions for improved face recognition," in *Proceedings of the Applied Imagery and Pattern Recognition Workshop*, 2005, pp. 100–105.

[7] P. N. Belhumeur and D. J. Kriegman, "What is the set of images of an object under all possible lighting conditions," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 28, 1998, pp. 270–277.

[8] V. Blanz and T. Vetter, "Face recognition based on fitting a 3d morphable model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1063–1074, 2003.

[9] J.-T. Chien and J.-C. Chen, "Recursive bayesian regression modeling and learning," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 2, 2007, pp. 557–560.

[10] B. Moghaddam, T. Jebara, and A. Pentland, "Bayesian face recognition," *Pattern Recognition*, vol. 33, no. 11, pp. 1771–1782, 2000.

[11] Z. Zhang, X. Xu, and T. Huang, "Indecisive classifier," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2005.

[12] W.-S. Kang and J. Y. Choi, "Kernel machine for fast and incremental learning of face," in *Proceedings of the International Joint Conference SICE-ICASE*, 2006, pp. 1015–1019.

[13] K. Haynes, X. Liu, and W. Mio, "Recognition using rapid classification tree," in *Proceedings of the IEEE International Conference on Image Processing*, 2006, pp. 2753–2756.

[14] J. Chen, X. Chen, and W. Gao, "Resampling for face detection by self-adaptive genetic algorithm," in *Proceedings of the International Conference on Pattern Recognition*, vol. 3, 2004, pp. 822–825.

[15] Y. Hori, K. Shimizu, Y. Nakamura, and T. Kuroda, "A real-time multi face detection technique using positive-negative lines-of-face template," in *Proceedings of the International Conference on Pattern Recognition*, vol. 1, 2004, pp. 765–768.

[16] C.-P. Liao and J.-T. Chien, "Maximum confidence hidden markov modeling for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 606–616, 2008.

[17] A. V. Nefian and M. H. H. Iii, "Hidden markov models for face recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1998, pp. 2721–2724.

[18] T. Mu and A. K. Nandi, "Ekf based multiple parameter tuning system for a l2-svm classifier," in *Proceedings of the IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, 2006, pp. 229–233.

[19] J.-W. Hong, M.-J. Han, K.-T. Song, and F.-Y. Chang, "A fast learning algorithm for robotic emotion recognition," in *Proceedings of the International Symposium on Computational Intelligence in Robotics and Automation*, 2007, pp. 25–30.

[20] G. Guo, S. Z. Li, and K. Chan, "Face recognition by support vector machines," in *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*, 2000, pp. 196–201.

[21] B. Heisele, Y. P. Ho, and T. T. Poggio, "Face recognition with support vector machines: global versus component-based approach," in *Proceedings of the International Conference on Computer Vision*, 2001, pp. 688–694.

[22] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006.

[23] X. Zhu, "Semi-supervised learning literature survey," Computer Sciences, University of Wisconsin-Madison, Tech. Rep., 2005.

[24] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[25] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[26] H. Byun and S.-W. Lee, "Applications of support vector machines for pattern recognition: A survey," *Lecture Notes in Computer Science*, vol. 2388, pp. 571–591, 2002.

[27] C. Campbell, "Kernel methods: a survey of current techniques," *Elsevier Neurocomputing*, vol. 48, pp. 63–84, 2002.

[28] K. Benabdeslem and Y. Bennani, "Dendogram-based SVM for Multi-Class Classification," *Journal of Computing and Information Technology*, vol. 14, no. 4, pp. 283–289, 2006.

[29] T.-J. Chin, K. Schindler, and D. Suter, "Incremental kernel svd for face recognition with image sets," in *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2006, pp. 461–466.

[30] F. Hui, W. Jin-wei, and L. Si-wei, "Dynamic grouping multi-classes face recognition method," in *Proceedings of the 6th International Conference on Signal Processing*, vol. 2, 2002, pp. 1259–1262.

[31] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Advances in Neural Information Processing Systems*, 2000, pp. 409–415.

[32] G. Fung and O. L. Mangasarian, "Incremental support vector machine classification," in *Proceedings of the Second SIAM International Conference on Data Mining*, 2002, pp. 247–260.

[33] W. Emara and M. M. Kantardzic, "An approach for incremental semi-supervised svm." in *Proceedings of IEEE Conference on International Conference on Data Mining Workshops*, 2007, pp. 539–544.

[34] J. Luo, A. Pronobis, B. Caputo, and P. Jensfelt, "Incremental learning for place recognition in dynamic environments," in *Proceedings of 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 721–728.

[35] G. Cauwenberghs and T. Poggio, "Incremental and decremental support vector machine learning," in *Advances in Neural Information Processing Systems*, 2000, pp. 409–415.

[36] AT&T Laboratories, Cambridge. The ORL Database of Faces. [Online]. Available: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html