

Structured Principal Component Analysis

Kristin M. Branson and Sameer Agarwal
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92193-0114

Abstract

Many tasks involving high-dimensional data, such as face recognition, suffer from the curse of dimensionality: the number of training samples required to accurately learn a classifier increases exponentially with the dimensionality of the data. Structured Principal Component Analysis (SPCA) reduces the dimensionality of the data while preserving its discriminative power. The algorithm finds clusters of similar features, where the similarity between features is measured using the class-conditional Chi-squared distance between the distributions of the features. As features in a cluster are similar and thus redundant, an entire cluster can be represented by a small number of principal components extracted from each cluster. We test the algorithm on two face recognition databases, the Ekman and Friesen Pictures of Facial Affect Database and the Yale Face Database, with encouraging results.

1. Introduction

Many tasks in machine learning and computer vision require learning a classifier from a small number of high-dimensional training samples. These tasks are particularly difficult because the potential complexity of a classifier increases exponentially with the dimensionality of the data.

For example, consider the task of image classification. The goal is to learn a simple classifier, say a perceptron, that will accurately classify novel images. The number of parameters to learn is more than the number of pixels in the image, and the number of samples required to accurately and confidently learn a perceptron is many more than the number of parameters. A standard dataset has over 10,000 pixels per image and 100 training images, thus an accurate perceptron cannot be learned from this data.

Fortunately, real-world data sets contain large amounts of redundancy, thus the data can be represented by small(er) sets of features. The pixels in an image are redundant, as images are highly structured. JPEG compression and image subsampling exploit redundancy to reduce dimensionality while retaining most of the image structure.

Two properties make dimensionality reduction for clas-

sification tasks efficient. First, classification tasks are restricted to small domains. For example, in face recognition, all data samples are images of faces. As all faces share the same structure and deviations between different face images are small, a few features can represent any face.

Second, dimensionality reduction with the ultimate goal of learning a classifier need only preserve the properties of the data relevant to the classification task. For example, if the classification task is face expression recognition, only the features relevant to describing the posed expression must be preserved, not the features describing the person's identity. Recent research has shown that features useful for identity recognition are orthogonal to those useful for expression recognition [4]. Including features encoding identity increases noise and complexity that will obfuscate classifier learning algorithms. Most dimensionality reduction techniques do not take advantage of this second property. These unsupervised algorithms instead ignore the class labels of the data and find the features that best represent all the properties of the data.

In this paper, we present a new supervised algorithm for dimensionality reduction, Structured Principal Component Analysis (SPCA), that preserves the class-conditional structure of the data. If two features are similar within every class, then given one feature, the second feature does not add much information useful for classifying the data. This means the dimensionality can be reduced by replacing clusters of features that are similar within every class with a small number of features. SPCA structures the features of the data into groups with high within-class similarity, then, for each cluster, performs Principal Component Analysis (PCA) on the data projected on the features in that cluster. SPCA thus finds a linear projection of the data that preserves class discriminability.

2. Related Work

SPCA was conceived with the faults of two classical algorithms in mind, Principal Component Analysis and Fisher's Linear Discriminant Analysis. In this section, we discuss these two algorithms and their flaws. In addition, we discuss Factor Analysis, which shares some ideas with SPCA.

Finally, we emphasize the dissimilarity of SPCA and mixture algorithms like Mixtures of Gaussians.

The classical unsupervised dimensionality reduction algorithm is PCA. PCA selects the orthonormal features among the linear combinations of the original features that maximize the variance of the projected data. While PCA is optimal in terms of its criterion, it is generally not optimal for classification tasks as PCA ignores the class labels. In the case of facial expression recognition, maximizing the variance of the projected data is not ideal, for face images vary more over identity than expression. Most features found will not be useful in expression classification.

Fisher’s Linear Discriminant Analysis (LDA), on the other hand, uses a supervised criterion to choose a set of orthonormal features from all linear transformations of the original features. The features are selected to minimize the variance of the data within each class while maximizing the variance of the means of each class of data. The standard tradeoff between these two goals is to maximize the quotient: the variance of the means of each class divided by the summed variance within each class. Note that LDA only depends on the mean and variance of the data. However, these two statistics are sufficient to describe the data only if the data is normally distributed. If this assumption does not hold, then it is not clear that LDA is optimizing the right criterion. For example, in many domains the distribution of a feature can be bimodal within a class. Another problem with LDA is that it can select at most $c - 1$ features, where c is the number of classes. In most cases, this is not enough to generalize to novel data samples.

The theme of Factor Analysis (FA) is similar to that of SPCA: if features are highly correlated, they can be represented by a few features in the directions of their correlation. FA represents each D -dimensional data sample \mathbf{x}_i by a d -dimensional *factor* \mathbf{z}_i such that \mathbf{z}_i probabilistically characterizes as much of the correlation between each dimension of \mathbf{x}_i as possible. Therefore, given \mathbf{z}_i , the dimensions of each \mathbf{x}_i are independent. There are two main differences between SPCA and FA that make SPCA a more powerful method. Most importantly, FA is *not* a supervised algorithm. SPCA is supervised because it uses the class-conditional proximity of the features’ distributions to measure similarity, as opposed to the unsupervised correlation between features used by FA. This results directly from the deep embedding of the similarity measure of FA in the algorithm, whereas in SPCA the similarity measure is explicit and flexible. Because it is unsupervised, FA finds features that are not useful for the ultimate classification task. Second, FA represents similar features by features in the direction of maximum covariance, whereas SPCA represents similar features by features in the direction of maximum variance. The maximum variance is a more robust measure, since if the variance of even one feature of a group in

a certain direction is large, this direction is most probably important for representing the data. Thus SPCA does not rely on the grouping of features being exact.

When one thinks of clustering, it is difficult not to think of clustering the data samples. However, Factor Analysis and SPCA group together features, not data samples. Mixture methods like Mixtures of Gaussians and Mixtures of Principal Component Analyzers find soft clusterings of the data samples, not the dimensions.

3. SPCA Algorithm Description

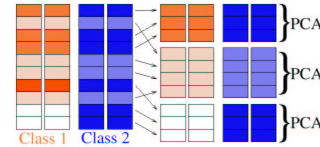


Figure 1: Illustration of the SPCA algorithm. On the left are four training data samples, two in each class. Each box represents a feature of the data. Color encodes feature values. SPCA groups together features that have similar distributions within classes. The clusters are shown on the left. PCA is performed on the data projected on each cluster of features separately.

SPCA finds features that preserve the class-conditional structure of the data. It clusters the features of the data into groups that have similar class-conditional distributions. No one cluster necessarily has more discriminative power than any other cluster. The hypothesis is that each cluster of features can be represented by just a few features, and that the collection of these few features from each cluster will preserve the discriminability of the data.

SPCA is an algorithmic framework because the pairwise similarity measure, the clustering algorithm used to group similar variables, and the method used to choose representative features from each cluster can all be varied. In this section, we describe the instantiation of the SPCA framework we implemented. First, the pairwise distance between each pair of features is measured by the class-conditional Chi-squared distance between the distributions of the features. Second, the Normalized Cut criterion is used to cluster the features. Finally, a few features are chosen to represent each cluster of features by PCA.

3.1 Feature Similarity Measure

A supervised measure of the similarity between two features, u and v , is the class-conditional distance between the distributions of the features. This is the weighted sum of the within-class distance between the distributions,

$$\sum_{c'=1}^c d(f_{u|c'}(\cdot), f_{v|c'}(\cdot))P(c'),$$

where c is the number of classes, d is an unsupervised function of distance between distributions, and $P(c')$ is the

probability of class c' . Because $f_{u|c'}$, $f_{v|c'}$, and $P(c')$ are unknown, they must be estimated from the training data. The distributions are estimated as the histogrammed data; $h_{u|c'}(i)$ is the number of samples of class c' such that the value of feature u falls within the interval i . The distance between the distributions is estimated as the distance between the histogrammed data, $d(h_{u|c'}(\cdot), h_{v|c'}(\cdot))$. The class probability is estimated by the Maximum Likelihood Estimate, $P(c') = n_{c'}/n$, where n is the total number of samples and $n_{c'}$ is the number of samples of class c' .

Any distance metric may be used for the unsupervised distance $d(\cdot, \cdot)$. We chose the Chi-squared distance metric:

$$d(h_{u|c'}(\cdot), h_{v|c'}(\cdot)) = \sum_{i=1}^k \frac{(h_{u|c'}(i) - h_{v|c'}(i))^2}{h_{u|c'}(i) + h_{v|c'}(i)},$$

where k is the number of intervals into which the data is divided. The Chi-squared distance was chosen because it is the standard, historically used metric to compare histogrammed data. The class-conditional pairwise distance between features u and v is therefore

$$\sum_{c'=1}^c \sum_{i=1}^k \frac{(h_{u|c'}(i) - h_{v|c'}(i))^2}{h_{u|c'}(i) + h_{v|c'}(i)} P(c').$$

3.2 Clustering Using Normalized Cut

SPCA uses the Normalized Cut algorithm to cluster the features so that features in the same cluster are similar, while features in different clusters are dissimilar [10]. Thus, SPCA clusters the features so that intra-cluster affinity is maximized while inter-cluster affinity is minimized, where affinity is group similarity. The similarity between features u and v is inversely proportional to the distance between them, $d(u, v)$: $W(u, v) = e^{-d(u, v)^2/\sigma}$ (σ is a constant that describes what distances are considered far). The inter-cluster affinity between clusters S_1 and S_2 is:

$$\text{Aff}(S_1, S_2) = \sum_{u \in S_1} \sum_{v \in S_2} W(u, v).$$

Similarly, the intra-cluster affinity of cluster S is:

$$\text{Aff}(S, S) = \sum_{u, v \in S} W(u, v).$$

The criterion function minimized by Normalized Cut is:

$$NCut(S_1, S_2) = \frac{\text{Aff}(S_1, S_2)}{\text{Aff}(S_1, S_1 \cup S_2)} + \frac{\text{Aff}(S_2, S_2)}{\text{Aff}(S_2, S_1 \cup S_2)}.$$

This quantity increases with inter-cluster affinity and decreases with intra-cluster affinity.

The membership vector, \mathbf{y} , that indicates which cluster each feature should be in would ideally be discrete valued, with a single value for each class. Finding the optimal

discrete-valued membership vector is an NP-hard problem. However, \mathbf{y} can be approximated by solving a generalized eigenvector problem, $W\mathbf{y} = \lambda D\mathbf{y}$. The pairwise affinity matrix, W , is a $N \times N$ matrix, where N is the original number of features in the data. Each element of the affinity matrix is the pairwise similarity $W(u, v)$ between two features, u and v . The degree matrix, D , is a diagonal matrix in which each diagonal element represents the total similarity of a feature to all other features. That is, $D(u, u) = \sum_{v=1}^N W(u, v)$ [11]. The vector \mathbf{y} is thresholded to determine which features are members of the same cluster.

The above formulation can be extended to a k -partitioning of the graph by using additional eigenvectors [9]. We do so by stacking the 2^{nd} to the k^{th} eigenvectors columnwise, normalizing the rows of the resulting matrix, and performing k -means clustering on them.

Given that our data is high-dimensional, solving the eigenvector problem is a computationally intensive task. However, our high dimensional data is highly redundant, i.e. there are a large number of features in our data that are similar to each other, implying that a number of rows of our weight matrix W are similar to each other. Having made this observation, we approximate the eigenvector decomposition by solving the problem for a random sample from the data and extrapolating the resulting eigenvectors to the full dataset. This is known as the Nyström approximation. The original eigenvector problem has complexity $O(D^3)$ in the dimensionality of the data. Using the Nyström approximation we can compute the eigenvalue decomposition in $O(s^3D)$, where s is the number of samples used. Empirical evidence shows that for data with a clear clustering structure, a fairly small number of samples can be used to approximate the eigenvectors to a small error [7].

3.3 Representation of Each Cluster

SPCA clusters the features of the data into groups that, because of their high affinity, can be represented by a small number of components to reduce dimensionality. As illustrated in Figure 1, if features u_1, \dots, u_m are grouped into one cluster, then a few features are chosen based on the data samples $\mathbf{X}_1 = (x_{1,u_1}, \dots, x_{1,u_m})^T$ through $\mathbf{X}_n = (x_{n,u_1}, \dots, x_{n,u_m})^T$, where $\mathbf{x}_1, \dots, \mathbf{x}_n$ are the training data samples. This is repeated for each cluster.

The concise representation closest to the actual data in each cluster is the top principal components of the data. PCA chooses the components that minimize the sum-squared distance between the projected data and the original data. These components are the eigenvectors of the sample covariance matrix, $\sum_i (\mathbf{X}_i - \mu)(\mathbf{X}_i - \mu)^T$, in order of the corresponding eigenvalue. Thus, there are two parameters in SPCA: the number of clusters and the number of features extracted from each cluster. As in PCA, the mean of the

data samples in each cluster is not represented in SPCA.

4 Experiments

The SPCA algorithm was compared to PCA and LDA on three sets of data. The first set is a synthetic set designed to demonstrate the weaknesses of PCA and LDA. The second set is the Ekman and Friesen POFA database, with the task of expression recognition. The third set is the Yale Face database, with the task of identity recognition. SPCA achieves 100% accuracy on the synthetic data, compared to SPCA and LDA which did no better than chance. SPCA also outperforms PCA and LDA on the POFA database. SPCA outperforms PCA on the Yale database and has similar performance to LDA.

4.1 Synthetic Data

PCA and LDA both have weaknesses that limit their effectiveness. If a feature has high variance but is uncorrelated with the class labels, PCA will highly represent this feature because of its variance, neglecting features with smaller variance but more correlated with the classification of the data. On the other hand, LDA assumes the class-conditional distribution of the data over each feature is normal. Suppose this assumption is false, for instance if a feature’s data for one class is bimodally distributed and for another class is normally distributed. This is the case for the pixels in the smiles (which may or may not show teeth) of happy faces versus pixels in the mouths of sad faces. As LDA chooses the components that separate the class means as much as possible, it will choose to offset the means of the bimodal and normal distributions. This could result in one of the modes of the bimodal distribution being projected to nearly the same value as the mean of the normal distribution.

Class-Conditional Distributions of Features

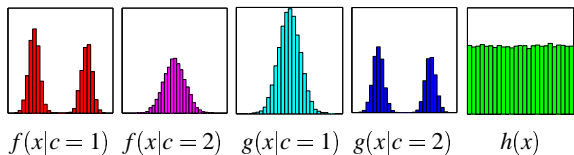


Figure 2: Distributions of the features of the synthesized data.

With these limitations in mind, we synthesized 100 training and 100 test samples, all i.i.d. Each sample has 1000 features with three possible distributions, $f(x|c)$, $g(x|c)$, and $h(x|c)$. Only the features with distribution f or g are useful in classification. These distributions are shown in Figure 2. $f(x|c=1)$ and $g(x|c=2)$ are bimodal distributions, with modes ± 0.5 and a standard deviation of 1. $f(x|c=2)$ and $g(x|c=1)$ are normal distribution with mean

0 and standard deviation 1. $h(x)$ is uniformly distributed between 0 and 1. 100 features have distribution f , 100 features have distribution g , and 800 features have distribution h . The optimal dimensionality reduction technique for this data set would ignore all 800 features of distribution h and use any of the features of distribution f or g .

Figure 3(a-d) shows the projection of the data on the features chosen by SPCA, LDA, and PCA. When grouping the data into three clusters, SPCA put all but two of the features with distribution f in one cluster, all but one of the features with distribution g in the second cluster, and all the rest of the features in the third cluster. Thus the first and second principal components generated by SPCA are useful in discriminating the data, while the third is not. SPCA performs equally well when only two clusters of the features are found.

As hypothesized, LDA was not able to separate the test data. It was able to find a projection to separate the training data, but this projection relied heavily on the features of distribution h which are not correlated with the classification. Thus, when generalizing to the test data, LDA fails.

PCA was distracted by the 800 features of distribution h that were not correlated with the classification, and thus was unable to separate the training and test data.

In fact, SPCA performs well while the other two algorithms fail on data in which the separation between the modes of the bimodal distribution is small. For separations greater than 0.1, SPCA achieves 100% accuracy using a nearest-neighbor classifier. No matter how small the separation, LDA and PCA are not able to separate the data, despite the distributions approaching a normal distribution, as shown in Figure 3(e).

These experiments on the synthetic data set show that SPCA is robust to features that are uncorrelated with classification, unlike PCA. They also show that SPCA is robust to non-normal distributions of the data, unlike LDA.

4.2 The Ekman and Friesen POFA Database

SPCA, PCA, and LDA were tested on the Ekman and Friesen Database of Pictures of Facial Affect [6]. This data set includes 14 trained actors posing six expressions, plus neutral. There are 110 greyscale images in this data set, 96 of which are not neutral. Examples from are shown in Figure 4(a).

An expression classifier must generalize over identity and concentrate only on the expression in an image. A supervised algorithm would be able to find a more accurate and concise representation that is tailored to expression recognition, in comparison to PCA. However, PCA significantly outperforms LDA, by a margin of 10% accuracy. We hypothesized that this was partially due to the limited number of components LDA can extract ($6 - 1 = 5$). Even trying

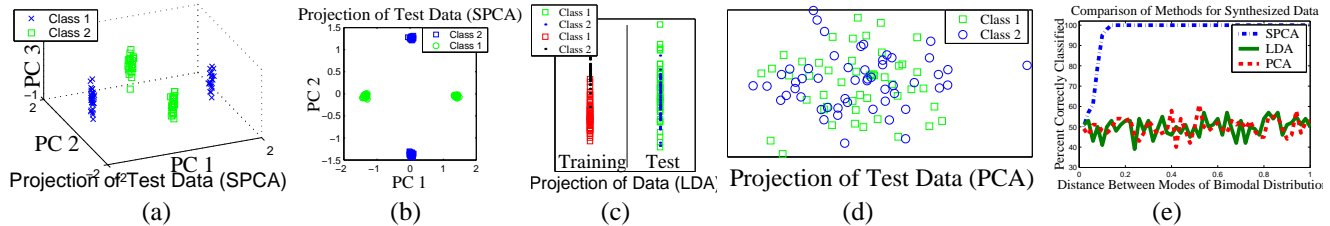


Figure 3: (a) Projection of the test data on the features chosen by SPCA, 3 clusters (b) 2 clusters (c) Projection of the training data and test data on the feature chosen by LDA. (d) Projection of the test data on the top two Principal Components chosen by PCA. (e) Results of SPCA, LDA, and PCA followed by a nearest neighbor classifier on classifying the synthetic data, with varying distance between the modes of the bimodal distributions.



Figure 4: (a) Example cropped and aligned images from the POFA database (b) Example full-face images from the Yale database (c) Example closely-cropped images from the Yale database.

different criterion functions which allow LDA to produce more features does not greatly improve LDA’s performance.

To compare SPCA with previous experiments in which PCA and LDA performed well, we perform the same image preprocessing. The images were aligned so that the eyes and the bottom of the top row of teeth were in the same position for all images, and cropped inside the contours of the face. Next, the images were subsampled and convolved with Gabor wavelet jets of 40 Gabor filters (five scales and eight orientations), resulting in a 40,600 dimensional vector. Finally, the outputs of the Gabor filters were z-scored (normalized so that the mean intensity value for each pixel is zero and the standard deviation is one). After preprocessing, the dimensionality of the data is reduced using PCA, LDA, or SPCA. A perceptron is learned from images of 12 of the actors, training is stopped at the best performance on a held out actor, and evaluated on a novel actor [5].

SPCA only finds clusters of features with high affinity, not necessarily important to classification. Clusters differ in number of features and correlation with the classification, yet the number of principal components extracted from each is equal. Thus, each cluster is weighted equally. For the POFA data set, we added an extra layer of PCA to weight the principal components extracted by SPCA by the amount of variance of the data projected on them.

This extra layer proved necessary when using a perceptron for classification, as a perceptron is greatly influenced by input variables that have small variance in the training data. For example, suppose a feature has a constant, low value for all the training data except for one of class c . The perceptron will find this feature useful in determining class c , and could weight its inputs to classify an example as class c if ever the value of this feature is high. If the inconsistent value for this one training sample is merely noise, the perceptron will mistake all test examples with an inconsistent

value for this variable as class c .

With an extra layer of PCA added, SPCA achieves 92.7% accuracy on this data set, compared to 90% accuracy for PCA (using 50 principal components), and 79.3% accuracy achieved by LDA. These are the optimal results obtained by SPCA, PCA and LDA. These results are impressive because 92.7% is 0.3% less than the accuracy humans achieve on this dataset.

SPCA proved to be relatively insensitive to the number of clusters and the number of principal components extracted from each cluster. Figure 5(a) and (b) show the results of varying these parameters. While SPCA performs better with 30 clusters than 20 clusters, the classification error difference is small, 2%. In addition, using 30 clusters, optimal results are obtained extracting two and four principal components from each cluster, and extracting three principal components is only 1% worse in classification error.

4.3 The Yale Face Database

The Yale Database [2] consists of images of 15 actors under 11 different conditions, including different lighting, facial expressions, and occlusion effects. Identity recognition is difficult, particularly for PCA, because the classifier must generalize over all these distractions [1]. This dataset was created with LDA in mind, thus LDA performs extremely well while PCA performs poorly in this experiment.

Two experiments were performed, one in which the images were cropped outside the face contour (full-face images) and one in which the images were cropped inside the face contour (closely-cropped images). Examples are shown in Figure 4. The preprocessing of this dataset is the same as that of the POFA dataset. After preprocessing, the dimensionality of the data is reduced using PCA, LDA, or SPCA. A perceptron is trained by backpropagation on 164

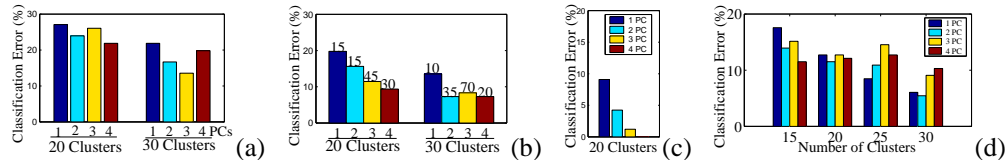


Figure 5: Comparison of parameter settings on POFA and Yale databases. (a) POFA, 20 and 30 clusters, without an extra layer of PCA (b) POFA, 20 and 30 clusters, with an extra layer of PCA. The numbers above each bar are the number of principal components extracted in the extra layer of PCA (c) Yale, Full-face images (d) Yale, Closely-Cropped images.

of the samples and tested on a novel image.

SPCA achieves 100% accuracy on the full-face samples, compared to LDA which obtains 99.4% classification accuracy and PCA which obtains 90% classification accuracy. On the closely-cropped samples, LDA outperforms SPCA. LDA achieves 97% classification accuracy, compared to SPCA with 94.6% accuracy and PCA with 76.4% accuracy. A comparison of the effects of the parameters for SPCA is shown in Figure 5(c) and (d).

5. Discussion

SPCA uses a supervised measure of similarity to cluster the features into groups of high intra-cluster affinity and low inter-cluster affinity. It extracts a small number of principal components from each cluster to represent the data. Experimentally, we have shown that the supervised measure of similarity allows SPCA to distinguish features that are correlated with the classification from those that are not. Because of this, SPCA outperforms PCA in all experiments. We have also shown that when the assumptions made by LDA do not hold, LDA performs very poorly. In these cases, we have experimentally shown that SPCA outperforms LDA. If the assumptions made by LDA do hold, then LDA performs optimally. In addition, we hypothesize that additional experimentation with non-aligned databases will show that SPCA is more robust than PCA and LDA to small translations and rotations in the images.

As stated earlier, SPCA is actually a versatile framework of algorithms. In the future, we hope to experiment with other instantiations, including different methods of representing the features of each cluster. Instead of selecting from the linear combinations of the features in a cluster, we could select directly from the features in the cluster. This would be useful in applications in which linear combinations of features are meaningless. We would also like to try mutual information measures of similarity, like the Kullback-Liebler distance.

Finally, we believe that the distance measure chosen for SPCA could be applied to LDA. In such an algorithm, the data would be projected onto the feature space which maximizes the Chi-squared distance between the distributions of the data of each class and minimizes the Chi-squared distance of the distributions within each class.

Acknowledgments

We would like to thank Serge Belongie, Gary Cottrell and GURU, Sanjoy Dasgupta, Virginia de Sa, Charles Elkan, and Bianca Zadrozny for helpful discussion and advice.

References

- [1] Belhumeur, P. N., Hespanha, J., and Kriegman, D. J., "Eigenfaces using class specific linear projection," *European Conference of Computer Vision*, Vol. 1, pp 45-58, 1996.
- [2] Belhumeur, P. N. and Kriegman, D. J., *The Yale Face Database*, 1997.
- [3] Bishop, C. M., *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
- [4] Cottrell, G. W., Branson, K. M., and Calder, A. J., "Do expression and identity need separate representations?," *Proceedings of the 24th Annual Conference of the Cognitive Science Society*, Fairfax, Virginia, pp 283-243, 2002.
- [5] Dailey, M. N., Cottrell, G. W., and Adolphs, R., "A Six-Unit Network Is All You Need to Discover Happiness," *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, Mahwah, New Jersey, 2000.
- [6] Ekman, P. and Friesen, W., *Pictures of Facial Affect*, Consulting Psychologists, Palo Alto, 1976.
- [7] Fowlkes, C., Belongie, S., and Malik, J., "Efficient Spatiotemporal Grouping Using the Nyström Method," *Computer Vision and Pattern Recognition*, Vol. 1, pp. 723-730, 2001.
- [8] Ghahramani, Z. and Hinton, G. E., "The EM Algorithm for Mixtures of Factor Analyzers," *Technical Report CRG-TR-96-1*, University of Toronto, 1996.
- [9] Ng, A. Y., Jordan, M. I., and Weiss, Y., "On Spectral Clustering: Analysis and an Algorithm," *NIPS* Vol. 14, 2002.
- [10] Shi, J. and Malik, J., "Normalized Cuts and Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22(8), pp 888-905, 2000.
- [11] Weiss, Y., "Segmentation using eigenvectors: A unifying view," *International Conference on Computer Vision*, Volume 2, pp 975-982, 1999.