To appear in Proceedings IEEE Workshop on Neural Networks for Signal Processing '95

LEARNING A DISTRIBUTION-BASED FACE MODEL FOR HUMAN FACE DETECTION

Kah-Kay Sung and Tomaso Poggio

Artificial Intelligence Laboratory
and
Center of Biological and Computational Learning
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Abstract

We present a distribution-based modeling cum example-based learning approach for detecting human faces in cluttered scenes. The distribution-based model captures complex variations in human face patterns that cannot be adequately described by classical pictorial template-based matching techniques or geometric model-based pattern recognition schemes. We also show how explicitly modeling the distribution of certain "face-like" non-face patterns can help improve classification results.

1 Introduction

Finding human faces automatically in a cluttered image is an important first step to a fully automatic face recognition system. It also has many potential applications ranging from surveillance and census systems to human-computer interfaces. Human face detection is difficult because there can be huge variations in the appearance of face patterns. Because many of these variations are difficult to parameterize, traditional fixed template pattern matching techniques [2] [3] and geometrical model-based object recognition approaches [1] tend to perform inadequately for detecting faces. Some non-parametric approaches [6] [4] have been recently proposed for representing and detecting face patterns, but so far, they have only been successfully demonstrated on images with little background clutter.

This paper describes an example-based learning approach for finding un-occluded vertical frontal views of human faces in cluttered scenes. To capture the full range of permissible variations in face patterns, the approach synthesizes a distribution-based model of frontal face views from an example database of face images. In order to perform pattern matching with the model, it learns a set of classification thresholds and parameters for separating "face" and "non-face" patterns, based on a set of distance measurements between the test pattern and the model.

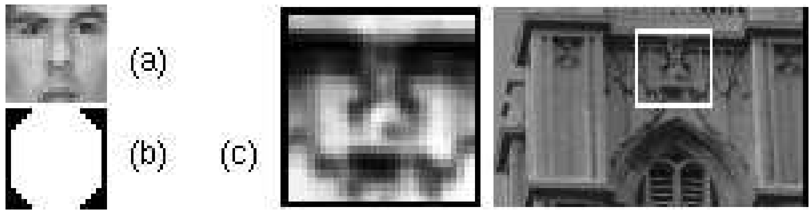

Figure 1: **(a):** A "canonical" face pattern. **(b):** A 19×19 mask for eliminating near-boundary pixels of canonical face patterns. **(c):** An example of a naturally occuring pattern that resembles a human face when viewed in isolation, but is clearly not a face when viewed in the context of its environment.

2 System Overview and Approach

Our approach finds faces by searching the image for square patches of the human face, whose upper boundary lies just above the eyes and whose lower edge falls just below the mouth (see Figure 1(a)). We shall henceforth refer to square patterns of this general structure as "canonical face patterns". The search for these face-like window patterns is done over multiple scales. At each scale, the image is divided into overlapping square windows of the current size. The system then attempts to classify the enclosed image pattern under each window as being either "a face" or "not a face" of the current scale, by matching the enclosed pattern with our face model.

Clearly, the most critical part of our system is the algorithm for classifying window patterns as "faces" or "non-faces". Section 3 describes the distribution-based model for representing face patterns. Section 4 describes the distance measurements for matching new window patterns to the model, and the decision procedure for classifying window patterns based on their distance measurements. In Section 5, we analyze our face detection technique and evaluate its performance.

3 A Distribution-based Face Model

Our distribution-based modeling scheme tries to represent canonical faces as the set of all masked 19×19 pixel patterns that are canonical face views. Suppose we apply the mask of Figure 1(b) to each 19×19 image and treat each unmasked pixel as a vector dimension, then the class of all masked 19×19 images forms a vector space whose dimensionality equals the number of unmasked image pixels. Each masked 19×19 image pattern maps to a specific vector space location, and the set of all 19×19 pixel canonical face patterns maps to a fixed volume in this multi-dimensional vector space. So, in theory, one can model the class of all canonical face views by identifying the portion of this multi-dimensional vector space that corresponds to canonical face patterns, and representing the region in some tractable fashion.

Figure 2 explains how we *approximate* the volume of canonical face patterns with limited data. Basically, we use a reasonably large example database of 19×19 canonical face patterns to obtain a coarse but fairly reliable representation of the actual canonical face manifold. We also use a carefully chosen database of non-face patterns to help refine the manifold representation by explicitly carving out regions around the "face" sample distribution that do not correspond to canonical face views. We shall explain how we synthesize our special database of "non-face" patterns in Section 3.2.

3.1 Modeling the Distribution of "Face" Patterns

We use a database of 4150 masked canonical "face" patterns to infer the volume of face views in our multi-dimensional image vector space. The database consists of 1067 real face patterns and 3083 *virtual* [5] face patterns, artificially generated from the real patterns via some simple affine transformations. The virtual patterns help ensure that our final database contains a reasonably dense and representative sample of canonical face patterns.

We approximate the "face" pattern distribution in a piecewise-smooth fashion using 6 multi-dimensional Gaussian clusters. Each Gaussian cluster consists of a centroid location and a full covariance matrix that describes the local data distribution around the centroid. The 6 clusters are obtained by performing *elliptical k-means* clustering on the "face" sample distribution (see [7] for details). The *elliptical* clustering algorithm differs from the traditional *k-means* algorithm in that it fits full covariance Gaussian clusters instead of isotropic Gaussian clusters to the data distribution. It approximates the "face" data distribution more closely with the same number of clusters, because locally, the "face" data distribution can be several orders of magnitude more elongated in certain image vector space directions than others.

The piecewise-smooth modeling scheme serves two important functions. First, it generalizes by applying a prior smoothness assumption to the observed "face" data distribution. Second, it serves as a tractable scheme for representing an arbitrary data distribution by means of a few Gaussian basis functions.

3.2 Refining the "Face" Distribution by Modeling "Non-Face" Patterns

There are many naturally occuring "non-face" patterns in the real world that look like faces when viewed in isolation (see for example Figure 1(c)). Because we are coarsely representing the canonical face manifold with 6 Gaussian clusters, some of these face-like patterns may even be located nearer the "face" cluster centroids than some real "face" patterns. This may give rise to misclassification problems, because in general, we expect the opposite to

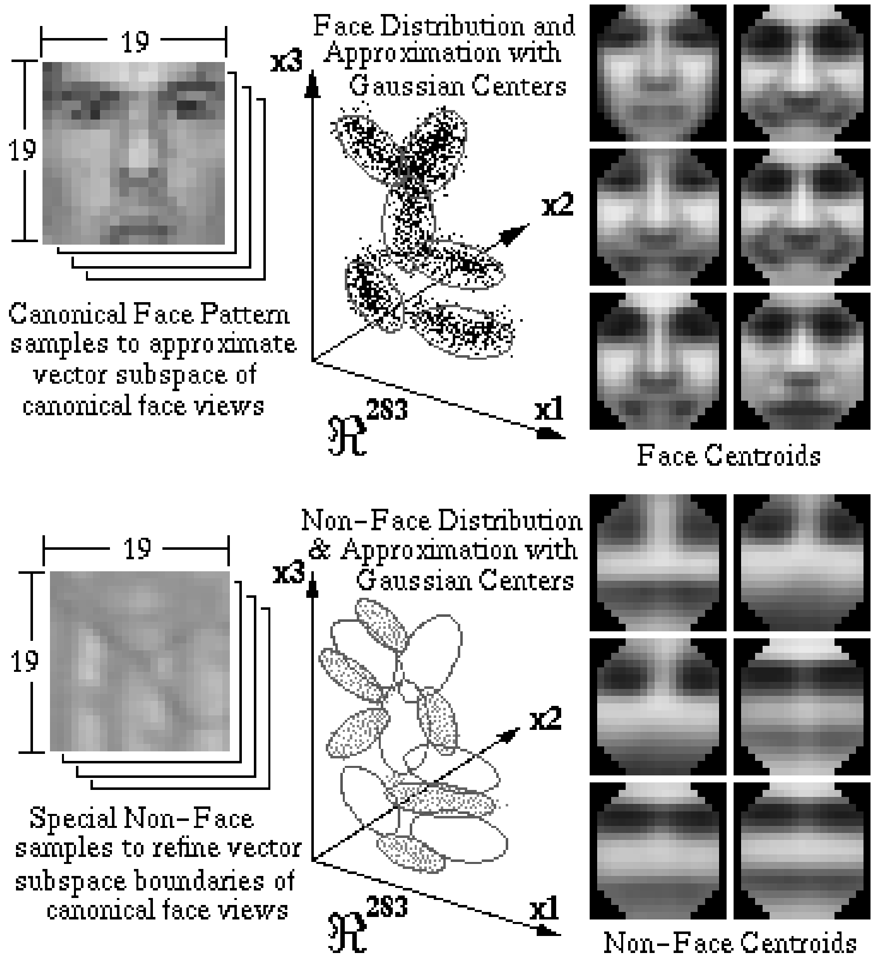

Figure 2: Our distribution-based canonical face model. **Top Row:** We use a representative sample of canonical face patterns to approximate the volume of canonical face views in a *masked* 19×19 pixel image vector space. We model the "face" sample distribution with 6 multi-dimensional Gaussian clusters. **Bottom Row:** We use a selection of non-face patterns to help refine the boundaries of our Gaussian mixture approximation. We model the "non-face" sample distribution with 6 Gaussian clusters. Our final model consists of 6 "face" clusters and 6 "non-face" clusters. Each cluster is defined by a centroid and a covariance matrix. The 12 centroids are shown on the right. **Note:** The two distribution plots are fictitious and are shown only to help with our explanation. The 12 centroid images are real.

be true, i.e. face patterns should lie nearer the "face" cluster centroids than non-face patterns.

We use our *elliptical k-means* algorithm to obtain 6 "non-face" clusters and their covariance matrices from a database of 6189 face-like patterns. The database was incrementally generated in a "boot-strap" fashion by first building a reduced version of our face detection system with only "face" clusters, and collecting all the *false positive* patterns it detects over a large set of natural images without faces.

Our final distribution-based model thus consists of 6 "face" clusters for coarsely approximating the canonical face pattern manifold in the image vector space, and 6 "non-face" clusters for refining the manifold by carving out non-face regions in the image vector space near the "face" clusters.

4 Matching and Classifying Patterns with the Model

To detect faces, our system resizes each candidate window pattern to 19×19 pixels and matches the resized pattern against our distribution-based face model to obtain a set of "difference" measurements. A trained classifier determines, based on the set of "difference" measurements, whether or not the test pattern is a frontal face view.

Each set of "difference" measurements is a vector of 12 distances between the test window pattern and the model's 12 cluster centroids in our multi-dimensional image vector space. One can interpret our vector of distances as the test pattern's displacement from 12 key reference location on the canonical face pattern manifold, and hence as a crude "difference" notion between the test pattern and the entire "canonical face" pattern class.

We use a 2-Value metric to encode the distance between the test pattern and a cluster centroid. The first distance value is a directionally dependent *Mahalanobis* distance between the test pattern and the cluster centroid, in a vector sub-space spanned by the cluster's 75 largest eigenvectors. This component computes a normalized pattern difference along the major directions of the local data distribution. It ignores pattern differences in the smaller eigenvector directions where the eigenvalue estimates are poor.

Let $\vec{x}$ be the column vector test pattern, $\vec{\mu}$ be the cluster centroid, E_{75} be a matrix with 75 columns, where column i is a unit vector in the direction of the cluster's i^{th} largest eigenvector, and W_{75} be a diagonal matrix of the corresponding 75 largest eigenvalues. The covariance matrix for the cluster's data distribution in the 75 dimensional sub-space is $\Sigma_{75} = (E_{75} W_{75} E_{75}^T)$, and the first distance value is:

$$\mathcal{D}_1(\vec{\mathrm{x}}, \vec{\mu}) = \frac{1}{2}(75 \ln 2\pi + \ln |\Sigma_{75}| + (\vec{\mathrm{x}} - \vec{\mu})^{\mathrm{T}} \Sigma_{75}^{-1} (\vec{\mathrm{x}} - \vec{\mu})).$$

The second distance component is a standard Euclidean distance between the test pattern $\vec{x}$ and its projection $\vec{x_p}$ in the 75 dimensional largest eigenvector sub-space. It is a robust directionally independent measure that accounts for pattern differences not captured by the first component.

$$\mathcal{D}_2(\vec{x}, \vec{\mu}) = ||(\vec{x} - \vec{x_p})|| = ||(I - E_{75}E_{75}^T)(\vec{x} - \vec{\mu})||.$$

We use a *multi-layer perceptron* (MLP) net to classify new window patterns as "faces" or "non-faces", based on their vector of distance measurements to the 12 cluster centroids. The net has 12 *pairs* of input terminals (for the 12 pairs of distance values) and one output node that returns a '1' for "face" patterns and a '0' otherwise. Our experiments show that the number of hidden units and network connectivity do not significantly affect the classifier's performance. We trained the net on a database of 4150 "face" patterns and 43166 "non-face" patterns. The non-face patterns include the 6189 face-like patterns used by our model for synthesizing the "non-face" distribution. The net was trained with a standard backpropagation learning algorithm until the output error stabilized at a very small value.

5 Results and Performance Analysis

Figure 3 shows some sample face detection results by our system. The system operates on window sizes of 19×19 pixels to 100×100 pixels at width increments of 120%. "Face" window patterns are marked with an appropriately sized dotted box in the output image. Many of the faces in Figure 3 are enclosed by multiple dotted boxes because the system has detected those faces either at a few different scales or at a few slightly offset window positions. The results show that the system (1) does not make many false positive errors (none in this case) even for fairly complex scenes, (2) detects faces successfully at different scales, and (3) detects real faces as well as hand-drawn faces. The system has a 96.3% face detection rate on a test database of 301 high quality CCD images of real people, and rarely makes any false detects, even on images with very cluttered background patterns.

We conducted the following experiment to determine the importance of "non-face" clusters in our distribution-based face model. We built two new systems: the first with 12 "face" clusters and no "non-face" clusters, and the second with only 6 "face" clusters. Table 1 compares the performance statistics of the two new systems with that of our original system. As expected, the original system with "non-face" clusters clearly out-performs the two systems without "non-face" clusters in terms of having a higher detection rate and fewer false positives. This suggests that the "non-face" distance measurements are in fact a very discriminative set of additional features for face pattern classification.

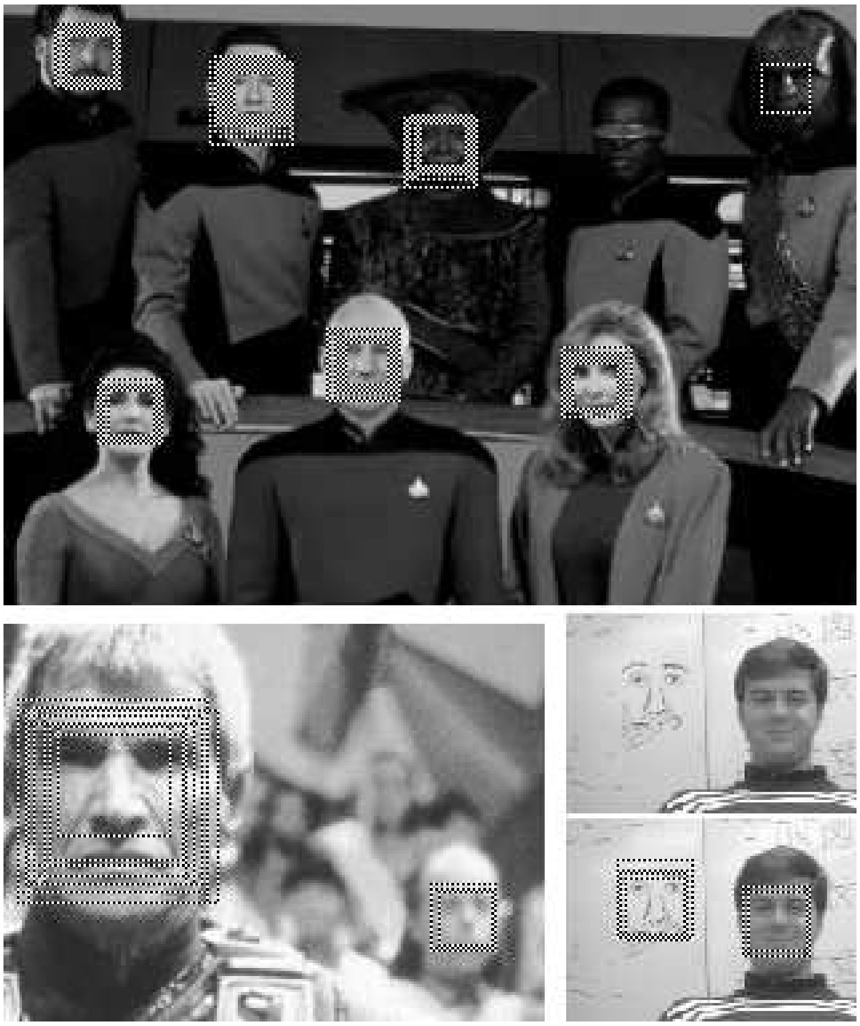

Figure 3: Some face detection results by our system. "Face" window patterns are marked with an appropriately sized dotted box in the output image. Many of the faces are enclosed by multiple dotted boxes because the system has detected those faces either at a few different scales or at a few slightly offset window positions. Notice that the current system does not detect Geordi's face in the top image. This is because Geordi's face differs significantly from the notion of a "typical" face pattern in our training database of face patterns — his eyes are totally occluded by an opaque metallic visor.

Classifier Architecture	Composition of Clusters in Model					
	6 Face & 6 Non-Face		12 Face		6 Face	
Multi-layer Perceptron	96.3%	3	85.3%	21	59.7%	17
Single Perceptron	96.7%	3	52.1%	6	66.6%	25

Table 1: Summary of performance figures for experiment on importance of "Non-Face" clusters. Detection rates versus number of false positives for different classifier architectures and composition of clusters in distribution-based model. The two numbers for each entry are: **Left:** detection rate on the test database of 301 high quality images. **Right:** total number of false positives for the test database.

6 Conclusion

We have successfully developed a distribution-based modeling cum example-based learning technique for representing and detecting frontal views of human faces in images. The distribution-based model captures pattern variations in face images that are difficult to parameterize using traditional pattern matching and object recognition techniques. We believe that the same distribution-based modeling cum example-based learning methodology can be easily extended to take on feature detection and pattern recognition tasks in other problem domains.

References

[1] W. E. L. Grimson and T. Lozano-Perez. Model-Based Recognition and Localization from Sparse Range Data. In A. Rosenfeld, editor, *Techniques for 3-D Machine Perception*. North-Holland, Amsterdam, 1985.

[2] B. Kumar, D. Casasent, and H. Murakami. Principal Component Imagery for Statistical Pattern Recognition Correlators. *Optical Engineering*, 21(1), Jan/Feb 1982.

[3] A. Mahalanobis, A. Forman, N. Day, M. Bower, and R. Cherry. Multi-Class SAR ATR using Shift-Invariant Correlation Filters. *Pattern Recognition*, 27(4):619–626, April 1994.

[4] A. Pentland, B. Moghaddam, and T. Starner. View-based and Modular Eigenspaces for Face Recognition. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, pages 84–91, June 1994.

[5] T. Poggio and T. Vetter. Recognition and Structure from One (2D) Model View: Observations on Prototypes, Object Classes, and Symmetries. A.I. Memo No. 1347, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1992.

[6] P. Sinha. Object Recognition via Image Invariants: A Case Study. In *Investigative Ophthalmology and Visual Science*, volume 35, pages 1735–1740, Sarasota, Florida, May 1994.

[7] K. Sung and T. Poggio. Example-based Learning for View-based Human Face Detection. Technical Report AIM–1521, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, December 1994.