To Appear in *Recent Progress in Computer Vision*, Springer-Verlag LNCS Series, 1995

# Finding Human Faces with a Gaussian Mixture Distribution-based Face Model

**Tomaso Poggio and Kah-Kay Sung**
Artificial Intelligence Laboratory
Massachusetts Institite of Technology
Cambridge, MA 02139, U.S.A.

**Abstract.** We present a distribution-based modeling scheme for representing and detecting human faces in cluttered scenes. A 2-Value metric is proposed for computing distance features between test patterns and the distribution-based model during classification. We present performance statistics of our overall system, and empirical results comparing the discriminative power of feature sets based on our 2-Value metric, versus similar feature sets based on other classical distribution dependent distance measures.

## 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 and geometrical model-based object recognition approaches tend to perform inadequately for detecting faces. Some non-parametric approaches [4] [3] 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 unoccluded vertical frontal views of human faces in cluttered scenes. The approach builds a distribution-based model of frontal face views to capture the full range of permissible variations in face patterns. In order to perform pattern matching with the model, we learn 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.

## 2 System Overview and Approach

At the top-most level, our approach finds faces by searching the image for square patches of the human face (henceforth refered to as "canonical face patterns"), whose upper boundary lies just above the eyes and whose lower edge falls just below the mouth (see Figure 1(a)). The search for these face-like window patterns

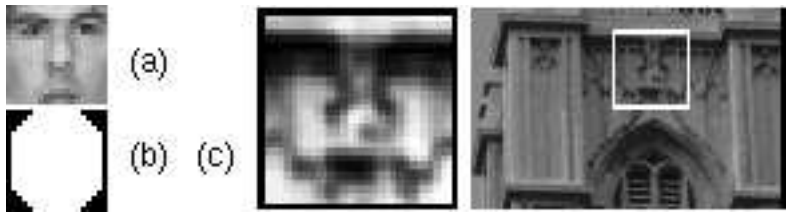


**Fig. 1. (a):** A "canonical" face pattern. **(b):** A $19 \times 19$ mask for eliminating near-boundary pixels of canonical face patterns. **(c):** An example of a naturally occuring pattern that resembles a human face when viewed in isolation. The same pattern is clearly not a face when viewed in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

is done at all image locations over multiple scales. At each image location and scale, the system classifies the local window pattern as being either "a face" or "not a face", based on its distance measurements to the face model.

Computational efficiency aside, clearly, the most critical part of our system is the algorithm for classifying window patterns as "faces" or "non-faces". The rest of this paper describes our distribution-based modeling scheme for representing and detecting canonical face patterns in images.

## 3 A Distribution-based Face Model

Our distribution-based modeling scheme tries to represent canonical faces as the set of all masked $19 \times 19$ pixel patterns that are canonical face views. Suppose we apply the mask of Figure 1(b) to each $19 \times 19$ image and treat each unmasked pixel as a vector dimension, then the class of all masked $19 \times 19$ images forms a vector space whose dimensionality equals the number of unmasked image pixels (283 in our case), and the set of all $19 \times 19$ pixel canonical face patterns maps to a manifold in this multi-dimensional vector space.

One can model the class of all canonical face views by recovering the manifold 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. We use a database of 4150 masked canonical "face" patterns to to obtain a coarse but fairly reliable representation of the actual canonical face manifold. We also use a carefully chosen database of 6189 non-face patterns to help refine the manifold boundaries 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.

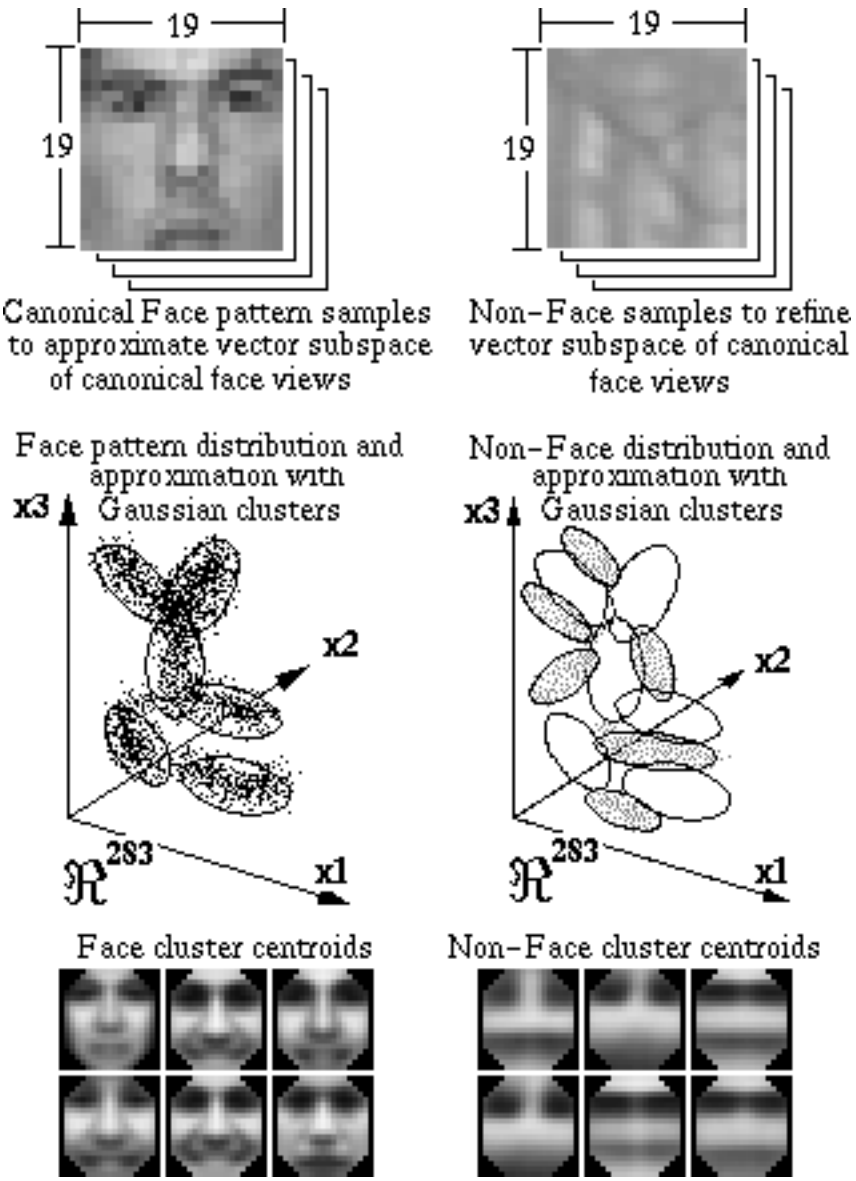


**Fig. 2.** Our distribution-based canonical face model. **Left Column:** We use a representative sample of canonical face patterns to approximate the volume of canonical face views in a *masked* $19 \times 19$ pixel image vector space. We model the "face" sample distribution with 6 multi-dimensional Gaussian clusters. **Right Column:** 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.

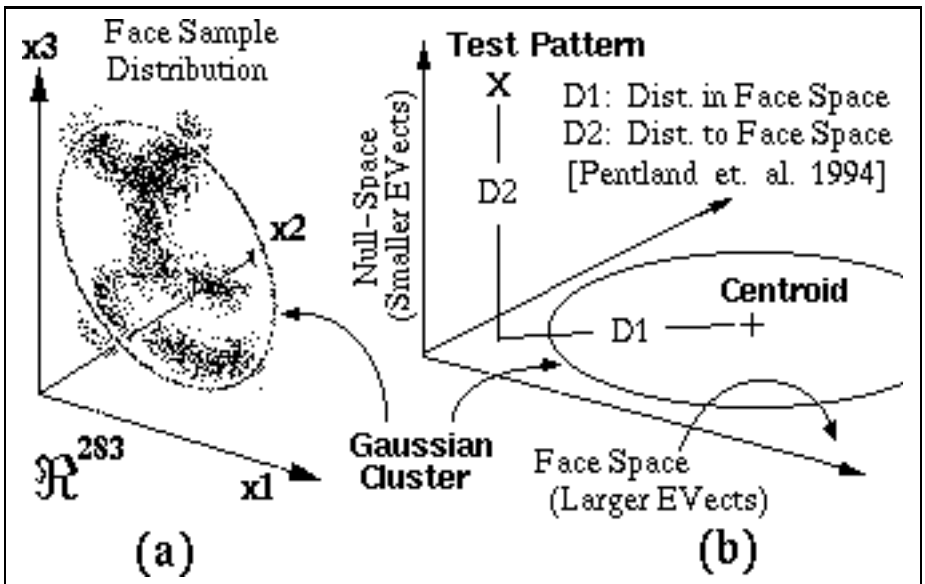


**Fig. 3. (a):** A single Gaussian cluster can be a very poor representation for an arbitrarily shaped "face" pattern distribution. **(b):** The two distance components we use in our scatter plots.

### 3.1 Modeling the "Face" Pattern Distribution with a Mixture of Gaussians

One can model the "face" pattern distribution by fitting the face data sample with a *single* multi-dimensional Gaussian cluster, consisting of a centroid location and a full covariance matrix. The following experiment shows that such a modeling scheme poorly describes the space of canonical face views. Using our face sample of 4150 patterns, we modeled the "face" distribution as a single multi-dimensional Gaussian cluster (see Figure 3(a)). For each face pattern in the sample, we resolved its displacement vector from the cluster centroid into two complementary components, similar to the *Distance in Face Space* and *Distance to Face Space* measures in [3] (see Figure 3(b)). We also resolved the displacement vectors of some non-face patterns from the "face" cluster centroid in the same fashion. Figure 4 shows that there is a significant amount of overlap between the face and non-face pattern distributions in this two distance component vector space, suggesting that a single Gaussian cluster representation does not separate the two pattern classes well. The distribution scatter plots are qualitatively very similar for different "Face Space" sizes.

Our approach approximates the "face" pattern distribution in a piecewise-smooth fashion using a few multi-dimensional Gaussian clusters (6 in our case). This model is reasonable as long as the actual face pattern distribution is *locally* linear, even though its *global* shape may be arbitrarily complex. The 6 clusters are

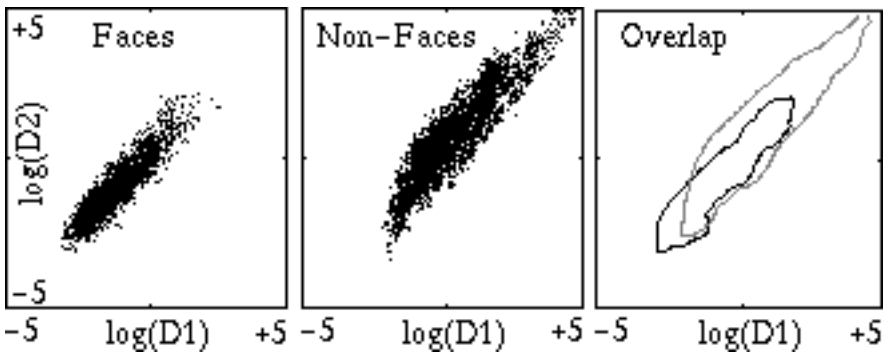


**Fig. 4.** Scatter plots to show that a single Gaussian cluster approximation poorly describes the space of canonical face views.

obtained by fitting 6 full covariance Gaussians to the "face" sample distribution using an *elliptical k-means* clustering algorithm (see [5] for details).

The piecewise-smooth modeling scheme serves two important functions. First, it generalizes the notion of "canonical faces" 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 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, one expects the opposite to be true, i.e. face patterns should lie nearer the "face" cluster centroids than non-face patterns.

To avoid confusion, 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 $19 \times 19$ 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 locations 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 a test pattern and each 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. Let $\mathbf{x}$ be the column vector test pattern, $\boldsymbol{\mu}$ be the cluster centroid, $E_{75}$ be a 75 column matrix, 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:

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

The second distance component is a standard Euclidean distance between the test pattern $\mathbf{x}$ and its projection $\mathbf{x_p}$ 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.

$$\mathcal{D}_2(\mathbf{x}, \boldsymbol{\mu}) = ||(\mathbf{x} - \mathbf{x_p})|| = ||(I - E_{75} E_{75}^T)(\mathbf{x} - \boldsymbol{\mu})||. \quad (2)$$

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. The detailed architecture of the net does not significantly affect the classifier's performance. We trained the net on our database of 4150 "face" patterns and 43166 "non-face" patterns, including the 6189 face-like patterns used for modeling the "non-face" distribution.

## 5 An Interpretation of our 2-Value Distance Metric

There is an interesting relationship between our 2-Value distance metric and the standard Mahalanobis distance measure that arises from fitting a full-covariance multi-dimensional Gaussian probability density to a sample data distribution. For high dimensional vector spaces, modeling a distribution with a full covariance

Gaussian is often not feasible because one usually has too few data samples to recover the covariance matrix accurately. In a $d$ dimensional vector space, each full-covariance Gaussian requires $d$ parameters to define its centroid and another $d(d-1)/2$ parameters to define its covariance matrix. For $d = 283$, this amounts to 40469 parameters!

One can greatly reduce the number of Gaussian model parameters by preserving only a small number of "significant" eigenvectors in the covariance matrix. Recently, Hinton et. al. [1] have used such an approach to model the distribution of hand-printed digits for digit recognition. One can show that a $d$ dimensional Gaussian with $h$ principal components has a covariance matrix of only $h(2d-h+1)/2$ free parameters. This can easily be a tractable number of model parameters if $h$ is small.

In the case of modeling our face pattern distribution, the $h$ principal components span a sub-space that preserves the most prominent pixel-wise correlations among face patterns. To exploit these pixel-wise correlations for pattern classification, one a directionally weighted Mahalanobis distance between a test pattern and the Gaussian centroid within this $h$ dimensional sub-space, i.e. $\mathcal{D}_1$ of our 2-Value distance metric.

The orthogonal subspace spanned by the $d-h$ remaining Gaussian eigenvectors encodes pixel correlations that are less prominent, and possibly also less reliable due to the lack of training data. We regularize this subspace by assuming an isotropic Gaussian distribution of data samples. This isotropic distribution requires only 1 free parameter to describe its variance, and we use a directionally independent Euclidean measure as distance, i.e. $\mathcal{D}_2$ of our 2-Value distance metric.

We can thus view our 2-Value distance metric as two complementary components of a robust approximate Mahalanobis distance that one uses when there is insufficient data to accurately recover the full covariance matrix of a Gaussian model. The two components can be combined to form a single robust Mahalanobis distance measure $\mathcal{M}_h$ by taking their weighted sum:

$$\mathcal{M}_h = \mathcal{D}_1 + \frac{d-h}{\sum_{i=h+1}^{d} \lambda_i} \mathcal{D}_2, \tag{3}$$

where $\lambda_i$ is the $i^{\mathtt{th}}$ eigenvalue of the Gaussian model. In a very recent piece of work, Moghaddam and Pentland [2] discusses about an optimal estimate for high dimensional Gaussian densities, whose form is a product of two lower-dimensional marginal and independent Gaussian densities. This estimate is exactly a re-expression of Equation 3 as a probabilistic measure.

## 6 Results and Performance Analysis

Figures 5 and 6 show some sample 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 mutiple dotted boxes

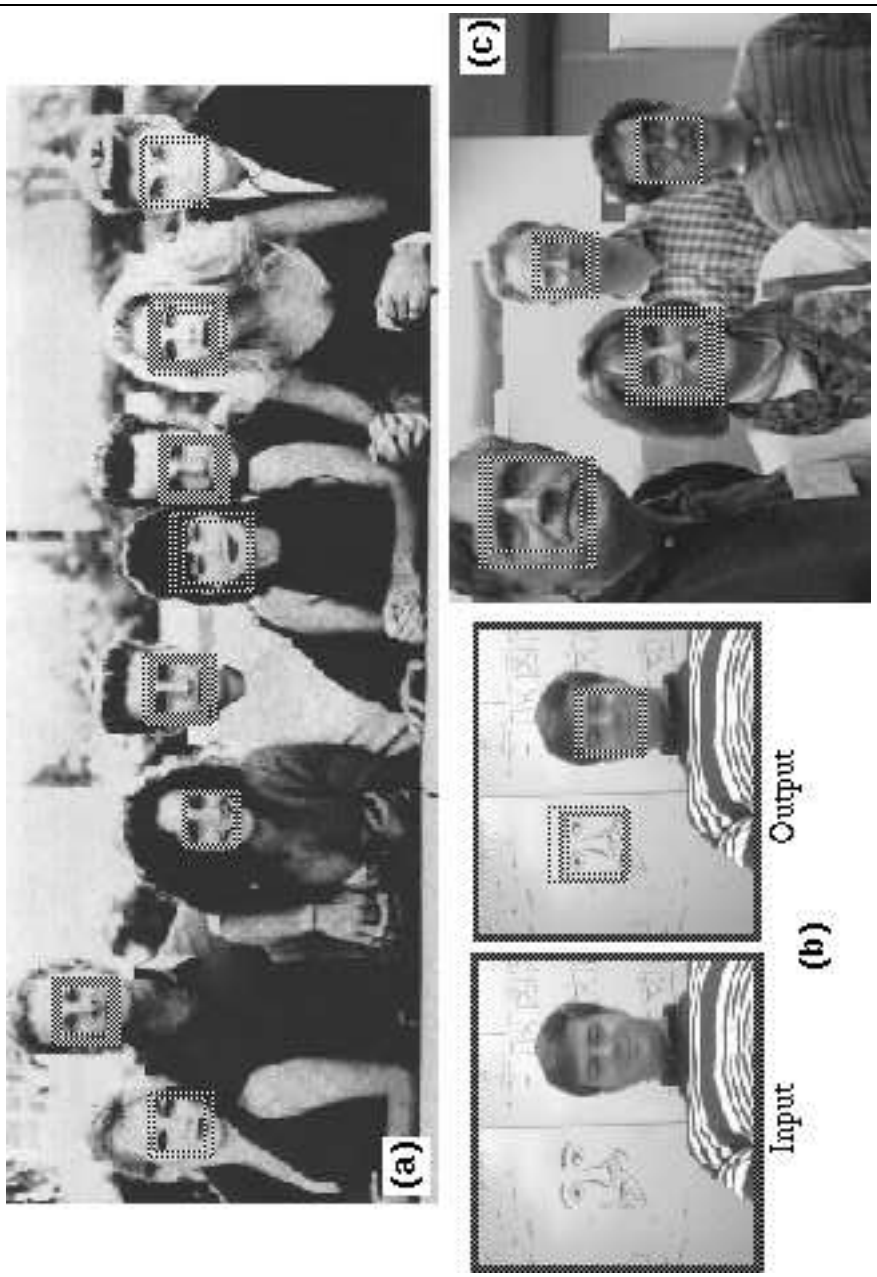


**Fig. 5.** Some face detection results by our system. **(a):** An image with many (8) frontal faces. The system finds all 8 faces successfully. **(b):** An input-output image pair. The same system detects real human faces and hand drawn faces. **(c):** The system finds faces successfully over multiple scales.

**Fig. 6.** More face detection results. The current system is trained to detect only frontal faces (Kirk and Uhura). It does not find side face profiles (Spock and McCoy) and partially occluded faces (Sulu).

because the system has detected those faces either at a few different scales or at a few slightly offset window positions. The system has a 96.3% face detection rate on a test database of 301 high quality CCD images of real people with only 3 false alarms.

We conducted the following experiment to compare the discriminative power of classification features based on our 2-Value metric, versus similar classification features based on (i) the standard Mahalanobis distance measure and (ii) its robust approximate form (equation 3). We built two new systems: The first system computes a vector of standard Mahalanobis distances between each test pattern and the 12 cluster centroids as input features to the MLP classifier, instead of the original vector of 2-Value distances. The second system computes a vector of robust Mahalanobis distances as the classifier input, with $h = 75$. We tested all three systems on a second more challenging database of 23 cluttered images with 149 face patterns, where all three systems are expected to produce more false alarms.

Table 1 compares the performance statistics of the two new systems with that of our original system. As expected, the original system with 2-Value distance features is strictly superior to the system using standard Mahalanobis distance features, in terms of having a higher detection rate and fewer false alarms. In comparing the original system with the one using robust Mahalanobis distance features, we felt that the original system produces results that are visually more

appealing.

| Distance Metric | Det. Rate | # False Det. |
|---|---|---|
| 2-Value Distance | 79.9% | 5 |
| Std. Mahalanobis | 42.6% | 5 |
| Robust Mahalanobis | 87.8% | 76 |

**Table 1.** Summary of performance figures for comparing three distance metrics as classifier input features in our distribution-based modeling face detection framework.

## 7 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 have also proposed a distribution dependent 2-Value metric for computing distance features between test patterns and the face model during classification. Empirically, the 2-Value metric leads to more appealing face-detection results than other classical distribution dependent distance measures, such as the standard Mahalanobis distance and its robust approximate form.

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. G. Hinton, M. Revow, and P. Dayan. Recognizing Handwritten Digits using Mixture of Linear Models. In D. Touretzky G. Tesauro and J. Alspector, editors, *Advances in Neural Information Processings Systems 7*, San Mateo, CA, 1995. Morgan Kaufman.
2. B. Moghaddam and A. Pentland. Probabilistic Visual Learning for Object Detection. In *Proceedings of the International Conference on Computer Vision*, pages 786–793, Cambridge, MA, June 1995.
3. 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.
4. 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.
5. 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.


This article was processed using the LaTeX macro package with LLNCS style