A distribution-based face/nonface classification technique

Son Lam Phung  
*University of Wollongong, phung@uow.edu.au*

Douglas Chai  
*Edith Cowan University*

Abdesselam Bouzerdoum  
*Edith Cowan University, bouzer@uow.edu.au*

Publication Details

A distribution-based face/nonface classification technique

Abstract
The core element of many existing approaches to face detection is the classification algorithm that determines if a sub-image of an input image contains a face pattern. In this paper, we present a novel and effective distribution-based face/non-face classification technique that detects frontal face patterns with possible in-plane rotation. A 15x15 input sub-image is first processed by a color filter, which verifies the presence of human skin color in the sub-image. Then, the intensity image is extracted from the identified skin color sub-image and converted into a vector in a high-dimensional space ($\mathbb{R}^{225}$). Principal component analysis is employed to reduce the dimension of this space to 20. In our approach, the distributions of face and non-face patterns in the $\mathbb{R}^{20}$ space are modeled using mixtures of Gaussians. The parameters of the Gaussian mixture models are determined through the use of the Expectation/Maximization (EM) algorithm. Finally, the classification of sub-images into face or non-face patterns is carried out through comparison of their estimated probability density functions. Experimental results have shown that the proposed technique is capable of performing highly accurate face/non-face classification.

Keywords
face, nonface, distribution, technique, classification

Disciplines
Engineering | Science and Technology Studies

Publication Details
A Distribution-Based Face/Non-Face Classification Technique

Son Lam Phung, Douglas Chai, and Abdessalem Bouzerdoum

Edith Cowan University
School of Engineering and Mathematics
100 Joondalup Drive, Joondalup WA 6027
Perth, AUSTRALIA

Abstract

The core element of many existing approaches to face detection is the classification algorithm that determines if a sub-image of an input image contains a face pattern. In this paper, we present a novel and effective distribution-based face/non-face classification technique that detects frontal face patterns with possible in-plane rotation. A 15x15 input sub-image is first processed by a color filter, which verifies the presence of human skin color in the sub-image. Then, the intensity image is extracted from the identified skin color sub-image and converted into a vector in a high-dimensional space (R^225). Principal component analysis (PCA) is employed to reduce the dimension of this space to 20. In our approach, the distributions of face and non-face patterns in the R^20 space are modeled using mixtures of Gaussians. The parameters of the Gaussian mixture models are determined through the use of the Expectation-Maximization (EM) algorithm. Finally, the classification of sub-images into face or non-face patterns is carried out through comparison of their estimated probability density functions. Experimental results have shown that the proposed technique is capable of performing highly accurate face/non-face classification.

1. Introduction

The research into face detection has attracted considerable attention in recent years. It deals with the problem of detecting the presence and the location of human faces in images. Many approaches to this problem involve a scanning process in which sub-images or windows of the input image are searched exhaustively for face patterns. They use various classification algorithms to determine whether or not a sub-image contains a face pattern. For example, Rowley et al. [15] proposed a neural network based approach that uses multilayer perceptrons to detect faces, and Fernand et al. [6] suggested a different neural network classifier that is based on the constrained generative model. Sung and Poggio [16], on the other hand, proposed a classifier that models the distribution of face and non-face with Gaussian clusters, found by using the elliptical k-means clustering technique. Osuna et al. [13] developed a classifier based on support vector machines, while Yang et al. [19] suggested a probabilistic method that uses a mixture of factor analyzers. For comprehensive reviews of these and many other color face detection techniques, readers are referred to the two recent survey papers, one written by Yang et al. [20] and the other by Hjelmas and Low [9].

In this paper, we propose a distribution-based technique for classifying image patterns into face and non-face. The probability density functions (pdfs) of face and non-face patterns are modeled with Gaussian mixtures. The parameters of the Gaussian mixtures are determined using the EM algorithm. The paper is organized as follows. Section 2 explains the proposed face/non-face classification technique. The distribution-based approach, which uses Gaussian mixtures, to the classification task is presented in Section 3. Experimental results along with some discussion can be found in Section 4, while conclusion is given in Section 5.

2. Face/Non-Face Classification

2.1 Overview

The proposed face/non-face classification technique is to work on a digital color image. The given image is divided into non-overlapping blocks of size 15x15, and these sub-images become the input data to the classifier. The 15x15 size is chosen after considering the trade-off between classification accuracy and computation load.

This classifier is currently being developed to detect not only frontal faces but also faces with some degrees of in-plane rotation. More details can be found in Section 4. A block diagram showing the three major components, or stages, of the classification technique is depicted in Figure 1. Their functions are as follows:

- In stage 1, a color filter is applied to the input sub-image in order to detect skin color pixels. The non-existence of face pattern is flagged if no or small amount of skin-color pixels has been detected. In this case, a non-face pattern is assumed and no further processing on this sub-image is required. If, on the other hand, the sub-image contains substantial skin-color pixels then the intensity values of the sub-image is extracted and fed as input to stage 2.
- In stage 2, more pre-processing steps are performed. The 15x15 input intensity image is first histogram equalized before being converted into a column vector in R^20 space. Principal component analysis (PCA) is used to reduce the dimension of this space. After applying PCA, each column vector in R^20 is represented by a feature vector w in R^20.
- In stage 3, the feature vector w is classified as face or non-face pattern by using a novel distribution-based algorithm. This algorithm is presented in Section 3.

As for stages 2 and 3, their respective functions are explained further in Sections 2.2 and 2.3.

2.2 Skin Color Detection

This component of the classification technique exploits the fact that human skin have distinct colors. The skin color filter used in the stage works in YCbCr color space with 8 bit per channel. The luminance value of a pixel (which corresponds to intensity) is stored in Y component, while the chrominance values are stored in Ch and Cr components. This color space is widely used in image/video coding standards such as JPEG, MPEG and H.266. Conversions between YCbCr and RGB color spaces can be done through a linear transformation, see [4] for an example.

It has been found that skin-color region can be identified by the presence of a certain set of chrominance values that is normally and intensively distributed in the YCbCr color space. It has also been proven that such model is robust against different types of skin color such as white, black, yellow, brown, etc. Further details of skin color model can be found in [2,3].

Here, a simple yet effective skin color filter is used to distinguish between skin color pixel and non skin color pixel. The filter identifies a pixel if the input sub-image as skin color if its chrominance components satisfy the following criteria:

\[ Ch \in [75, 135] \quad \text{and} \quad \text{Cr} \in [130, 180]. \]  

(1)

The skin color filter partitions the input sub-image P into two disjoint sets: \( P = P_{\text{face}} \cup P_{\text{non-face}} \). The input sub-image is considered as non-face if:

\[ \left| P_{\text{face}} \right| < \left| P_{\text{non-face}} \right| \]

(2)

where \( |P_A| \) denotes the number of elements in set \( P_A \) and \( \left| P_{\text{face}} \right| \) is a fixed threshold.

If the sub-image \( P \) passes this skin color test, its corresponding intensity sub-image is extracted and sent to stage 2. Here we use the luminance values of the sub-image, which we denoted by \( Y \), to represent the intensity sub-image.

Note that more sophisticated filtering techniques such as those that use neural networks [14], Gaussian [12], elliptical skin model fitting [1], and mixture of Gaussians [18] can be employed instead of the model of Eq. (1).

2.3 Data Reduction

In stage 2 of our approach, the intensity sub-image \( Y \) is histogram-equalized so as to reduce the effects of lighting variations. From the enhanced intensity sub-image, a vector \( x \) of 255 elements (ie 15x15) is formed by reading \( Y \) column-wise.

The pattern \( x \) has a relatively large dimension (\( D = 225 \)), and it contains some data that are not significant for classification purposes. Hence, we employ principal component analysis (PCA) to remove the irrelevant data and reduce the space dimension. In doing so, we cut down the computation load significantly. Note that to further reduce data, a mask that excludes boundary pixels from each pattern can be applied before classification—see [13, 15, 16] for further information.

132

Australian Journal of Intelligent Processing Systems

Volume 7, No.34

Spring/Summer 2000

Australian Journal of Intelligent Processing Systems
PCA works as follows. Let \( \{x_1, x_2, \ldots, x_n\} \) be a set of known face patterns in \( \mathbb{R}^D \) space. The PCA aims to find a set of \( k \) (typically \( k \ll D \)) orthogonal vectors, \( V = \{v_1, v_2, \ldots, v_k\} \), so that the distances (ie reconstruction errors) between \( x_i \) and their projections onto the subspace spanned by the \( k \) vectors are minimized. The steps for finding \( V \) are as follows:

- Compute the mean face vector:

  \[
  x = \frac{1}{N} \sum_{i=1}^{N} x_i
  \]

- Compute the covariance matrix:

  \[
  C = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - x)(x_i - x)^T
  \]

- Compute the set of all \( N \) eigenvectors of \( C \). A vector \( v \) is an eigenvector of \( C \) if there exists a scalar \( \lambda \) such that:

  \[
  Cv = \lambda v
  \]

  The scalar \( \lambda \) is called an eigenvalue of \( C \).

- Select \( k \) most significant eigenvectors, \( V = \{v_1, v_2, \ldots, v_k\} \), that correspond to the largest \( k \) eigenvalues of \( C \).

Once the significant eigenvectors (also termed as "eigenfaces" by Turk and Pentland [17]) are computed, each new pattern \( x \) is represented by the projection of its deviation from the mean face vector onto the subspace spanned by the column vectors of \( V \):

\[
 w = V^T(x - x_m)
\]

where \( w \) is a vector in \( k \)-dimensional space. In our design, we have set \( k \) to 20.

The problem of classifying the feature vector \( w \) into face or non-face class is dealt with in the third component of our classification technique, which is discussed in the next section.

3. Distribution-Based Classifier

The feature vector \( w \) for face and non-face is assumed to have arisen from two distinct, constrained but unknown distributions in \( \mathbb{R}^k \) space. Let \( p(w_{\text{face}}) \) and \( p(w_{\text{nonface}}) \) be the pdfs of face and non-face patterns, respectively. By applying the Bayes decision rule for maximum cost [84] to this two-class (ie face and non-face) classification problem, we classify the feature vector \( w \) as face if the following equation is satisfied:

\[
 \frac{p(w|\text{face})}{p(w|\text{nonface})} \geq \tau_f,
\]

where \( \tau_f \) is a fixed threshold, which is dependent on various classification costs and a priori probabilities of face and non-face. However, the value of \( \tau_f \) is often determined experimentally. The derivation of (7) is given in the Appendix.

A difficult problem remains, and that is to find the probability density functions \( p(w_{\text{face}}) \) and \( p(w_{\text{nonface}}) \). The common non-parametric approaches to pdf estimation such as histograms and kernel-based methods are not feasible in this situation due to the high dimension of the feature vector \( w \). We, therefore, propose a parametric approach for estimating \( p(w_{\text{face}}) \) and \( p(w_{\text{nonface}}) \) using mixtures of Gaussians.

3.1 Gaussian Mixture Modeling of Face PDF

The Gaussian mixture technique models the pdf of face patterns \( w \) as a linear combination of a set of \( G \) Gaussian components:

\[
 p(w|\text{face}) = \sum_{i=0}^{G} \frac{1}{N} \sum_{j=1}^{N} g(w; \theta_i)
\]

where \( \theta_i = \{\mu_i, \Sigma_i\} \) are the mixing factors and the parameter of the \( i \)-th component, respectively. The mixing factors must satisfy the following conditions:

\[
 \sum_{i=0}^{G} \pi_i = 1 \quad \text{and} \quad \pi_i \geq 0, i \in 1, G
\]

In our case, each component \( g(w; \theta_i) \) is a Gaussian, which is characterized by a mean vector \( \mu_i \) and a covariance matrix \( \Sigma_i \). Hence,

\[
 \theta_i = \{\mu_i, \Sigma_i\}
\]

\[
 g(w; \mu, \Sigma) = (2\pi)^{1/2} \left| \Sigma \right|^{-1/2} \exp \left[-\frac{1}{2}(w - \mu)^T \Sigma^{-1} (w - \mu) \right]
\]

Suppose we have a training set of face vectors \( \{w_1, w_2, \ldots, w_N\} \). The parameter set of the model \( \theta = \{\theta_0, \theta_1, \ldots, \theta_G\} \) for \( i = 1, G \) is estimated from the training vectors, and it must satisfy a condition known as maximum likelihood, which requires the following joint probability of occurrence of the training vectors to be maximized:

\[
 L = \prod_{i=1}^{N} p(w_i)
\]

A common method for parameter estimation using maximum-likelihood is the Expectation/Maximization (EM) algorithm.

3.1.1 EM Algorithm

The EM algorithm [5,7] starts with an initial estimate of the parameter set \( \theta \). This estimate is used to compute the pdf of \( x \) as in (8). The pdf of \( x \) is then used to compute a revised estimate of \( \phi \) (ie mixing factors \( \pi \), means \( \mu \), and covariance matrices \( \Sigma \)). This process continues for a fixed number of iterations until there is little change in \( \phi \), or until the following log-likelihood function exceeds a certain threshold:

\[
 \ln L = \sum_{i=1}^{N} \ln p(w_i)
\]

where the superscript \( t \) denotes the iteration number, the revised estimate of \( \phi \) at each stage is provided as follows:

- The revised estimate of the mixing factor for the \( i \)-th component is computed by

\[
 \pi_i^{t+1} = \frac{1}{N} \sum_{j=1}^{N} p(w_j | w_i)
\]

- The revised estimate of the mean for the \( i \)-th component is calculated by

\[
 \mu_i^{t+1} = \frac{1}{P} \sum_{j=1}^{P} p(w_j | w_i)
\]

- The revised estimate of the covariance matrix for the \( i \)-th component is given by

\[
 \Sigma_i^{t+1} = \frac{1}{P} \sum_{j=1}^{P} p(w_j | w_i) (w_j - \mu_i^{t+1})(w_j - \mu_i^{t+1})^T
\]

3.1.2 EM Initialization

The initial estimate for the EM algorithm can influence its convergence speed and final result. In our approach, Kohonen's Self-Organizing Map (SOM) algorithm [10] is applied to divide the training vector set \( \{w_j\} \) for \( j = 1, N \) into \( G \) clusters, \( C_i \), for \( i = 1, G \). The initial estimate of the parameter set \( \phi \) is then determined as follows:

\[
 \pi_i = \frac{N_i}{N}
\]

\[
 \mu_i = \frac{1}{N_i} \sum_{j=1}^{N_i} w_j
\]

\[
 \Sigma_i = \frac{1}{N_i} \sum_{j=1}^{N_i} (w_j - \mu_i)(w_j - \mu_i)^T
\]

where \( N_i \) is the size of cluster \( C_i \) and \( E(i) \) is the expectation operator.

3.2 Gaussian Mixture Modeling of Non-Face PDF

The pdf for non-face \( p(w_{\text{nonface}}) \) is also modeled as a mixture of Gaussians using essentially the same steps as described in Section 3.2. From a large number of non-face images, a set of non-face feature vectors \( w \) is obtained. This set is clustered using Kohonen's SOM algorithm to obtain an initial estimate of the parameters of the Gaussian mixture model. The EM algorithm is then performed to obtain the final estimate of the parameters. Note that the number of Gaussian components for non-face does not need to be the same as in \( p(w_{\text{face}}) \). Compared to the Gaussian mixture model for face, the model for non-face needs to be updated more frequently as new non-face patterns are presented to the classifier. This is due to the difficulty in finding a representative non-face pattern.

3.3 Face/Non-Face Classification

Once the parameters of the Gaussian mixture models are determined, face/non-face classification of the feature vector is carried out as follows. A feature vector is classified as face if it satisfies two conditions:

- Condition 1: \( p(w_{\text{face}}) > \tau_f \)
- Condition 2: \( p(w_{\text{nonface}}) > \tau_{\text{non}} \)

The first condition is the Bayes decision rule for maximum cost as mentioned earlier. The second condition is to help remove non-face patterns that pass the first condition.

4. Experiment Results

4.1 Training Procedure

The training set of face vectors was generated from the AR face database [11]. This database consists of more than 4,000 color frontal-face images of 126 people (70 men, 56 women) with various facial expressions that range from neutral, smile to anger. The images are of size 768x576. From this database, we extracted 1,000 face images, each of which has a size of 15x15. To generate sufficient training data as well as to detect face patterns that are influenced by in-plane rotation, each original face image was rotated by angles of \( \pm 30^\circ \) for \( i = 1, 10 \) to give extra 20 face images. The training set consisted of 21,000 face vectors.
PCA was performed on 1,000 face vectors and a set of 
k = 20 most significant eigenvectors was found. Each input 
pattern x was then represented by a feature vector w as 
computed in (6).

The Gaussian mixture model for face in R^d space consists of 
G = 5 components. The entire training set of 21,000 
feature vectors w was used to develop the model. The 
Gaussian mixture model for non-face also consists of G = 
5 components. A set of 21,000 non-face images randomly 
chosen from 100 training images was used to construct the 
non-face pdf. The various parameters of the face/non-face 
classifier are summarized in Table I.

Table I: Parameters of the face/non-face classifier.

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input sub-image with front-on in-plane rotation</td>
<td>Size: 15x15</td>
</tr>
<tr>
<td>Skin color range</td>
<td>Cb: [75, 135]</td>
</tr>
<tr>
<td>Skin color test threshold</td>
<td>t = 0.5</td>
</tr>
<tr>
<td>No. of eigenfaces</td>
<td>k = 20</td>
</tr>
<tr>
<td>No. of Gaussian components</td>
<td>G = 5 each</td>
</tr>
<tr>
<td>No. of face vectors for training</td>
<td>N = 21,000</td>
</tr>
<tr>
<td>Max. in-plane rotation allowed</td>
<td>g = ±50°</td>
</tr>
</tbody>
</table>

4.3 Results and Discussion

The classification results on the two test sets are provided in Table II.

They show that the proposed face/non-face classifier is capable of performing highly accurate classification. Most false rejections occur when the face is taken under views that are markedly different from the frontal views or under extreme lighting condition. A set of representative results of correct classification (true correct detection and rejection) are illustrated in Figures 2 and 3. Note that all samples of correct rejection as shown in Figure 3 have skin-color appearance and therefore they were not picked up as non-face patterns; however, they were detected as non-face in stage 3 of the classification process.

Table II: Face/non-face classifier testing results.

<table>
<thead>
<tr>
<th>Test set</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test set 1</td>
<td>650 faces</td>
</tr>
<tr>
<td>Test set 2</td>
<td>184,000 non-faces</td>
</tr>
<tr>
<td>Correct detection rate</td>
<td>89.2%</td>
</tr>
<tr>
<td>Correct rejection rate</td>
<td>98.9%</td>
</tr>
</tbody>
</table>

4.2 Testing Procedure

To overcome the difficulty of finding representative non-face images and also a test set that consists of a representative mixture of face and non-face images, the following testing strategy was adopted. The face/non-face classifier was evaluated on two different test sets. Test set 1 consisted of only face patterns and was used to estimate the correct detection rate. Test set 2 consisted of only non-face patterns and was used to estimate the correct rejection rate.

Test set 1 consisted of 650 color images. These images of size 15x15 were obtained from various sources including WWW and TV programs. The face images were mostly frontal with possible in-plane rotation and of people from diverse ethnicity ranging from Asian, African to European. Test set 2 consisted of 184,000 non-face images. These non-face patterns were obtained by using a program that randomly selected images of size 15x15 from 200 color images of various sizes. These color images, sized between 10^3 to 10^4 pixels, were obtained from both natural and man-made landscape photographs that contained no face patterns.

6. Acknowledgment

The authors wish to express sincere thanks to Dr Martinez at Purdue University for providing the AR face image database. The authors are also grateful to Mr Trivive at Edith Cowan University for taking part in preparing the images used in this work.

7. Appendix

For clarity, let C_f and C_n denote face class and non-face class, respectively. Let P(C_f|w) and P(C_n|w) be the a posteriori probabilities of C_f and C_n, respectively, i.e., probability of an observed pattern w belonging to a class.

Let c_f for t ≤ 1.2 denote the cost of classifying a pattern w into class C_f while it actually belongs to class C_n.

The cost of classifying a pattern w into class C_f is given by

R_f(w) = c_f * P(C_f|w) + c_n * P(C_n|w).

(22)

Similarly, the cost of classifying a pattern w into class C_n is given by

R_n(w) = c_f * P(C_f|w) + c_n * P(C_n|w).

(23)

The pattern w will be classified into the class that gives the minimum classification cost:

R_f(w) ≥ R_n(w) ⇒ w ∈ C_f
R_f(w) < R_n(w) ⇒ w ∈ C_n

(24)

By combining with (22) and (23), this decision rule can be rewritten as follows:

P(C_f|w) ≥ P(C_n|w) ⇒ w ∈ C_f
P(C_f|w) < P(C_n|w) ⇒ w ∈ C_n

(25)

According to Bayes theorem [8], the a posteriori probabilities P(C_i|w) can be expressed in terms of a priori probabilities P(C_i), class-conditional densities p(w|C_i) and unconditional probability p(w):

P(C_i|w) = P(w|C_i)P(C_i) / p(w).

(26)

P(C_i|w) = p(w|C_i)P(C_i) / p(w).

(27)

5. Conclusion

This paper presents a novel and accurate distribution-based 
classification technique for face and non-face patterns. The 
correct detection rate of 89.2% and the correct rejection 
rate of 98.9% were achieved. The classifier involves a skin 
color filter and PCA for dimension reduction of the input 
space. The key element of the classifier is the parametric 
estimation of probability density functions for face and 
non-face patterns using mixtures of Gaussians. The 
parameters of the estimation are found by using the EM 
algorithm. Face and non-face patterns are then classified 
by comparing the probability density functions.

The current classifier caters for the detection of frontal 
faces with some degree of in-plane rotation. We believe 
that the proposed approach is readily extendible to faces 
under different viewing angles. A possible method is 
to obtain a more comprehensive training set that covers 
greater viewing angles, and reconstruct the Gaussian 
models. This will be carried out in our future research work.
The decision rule in (25) becomes
\[
\begin{align*}
 p(w | C_1) & \leq \tau \Rightarrow w \in C_1, \\
 p(w | C_2) & \leq \tau \Rightarrow w \in C_2.
\end{align*}
\]
where \( \tau \) is a threshold that depends on the classification costs and a priori probabilities of face and non-face patterns, and it is defined as
\[
\tau = \frac{c_0 - c_1}{c_0 - c_2} \frac{P(C_1)}{P(C_2)}.
\]

8. References


9. Anti-correlation: A Diversity Promoting Mechanism in Ensemble Learning

R. I. (Bob) McKay 1 and Hussein A. Abbass 2
School of Computer Science, University of New South Wales, Australian Defence Force Academy Campus Northcott Drive, Canberra 2600, Australia,
(E-mail: rim@cs.adfa.edu.au, 2.h.abbass@adfa.edu.au)

Abstract

Anti-correlation has been used in training neural network ensembles. Negative correlation learning (NCL) is the issue of the art anti-correlation measure. We present an alternative anti-correlation measure, RTQRT-NCL, which shows significant improvements on our test examples for both artificial neural networks (ANN) and genetic programming (GP) learning machines. We analyze the behavior of the negative correlation measure and derive a theoretical explanation of the improvement in performance of RTQRT-NCL in larger ensembles.

Keywords: Anti-correlation, Artificial Neural Networks, committee learning, Ensemble learning, fitness sharing, genetic programming, diversity

1 Anti-Correlation Learning

Committee learning refers to a form of learning algorithm where a committee of learning machines is used to learn a task. The hope is that each member of the committee will specialize on a part of the task. In anti-correlation learning, a specific mechanism is incorporated in the learning mechanism to reduce correlation between the committee members, without unduly sacrificing the accuracy of prediction. The error function of each committee member needs, therefore, to have an additional penalty term which accomplishes the following:

1. It maximizes the distance between all networks; and therefore achieves a nice spread in the ensemble space.

2. It is dimensionally consistent with the error function.

3. It does not have a larger magnitude than the original error function; otherwise the effect will be dramatic (i.e., the networks will be so different that they contradict each other and the performance becomes chaotic).

The paper is organized as follows. In section 2, negative correlation learning is discussed, and an alternative, RTQRT-NCL, is introduced in subsection 2.3. We then use RTQRT-NCL for two learning machines: artificial neural networks in section 3 and genetic programming in section 4. The results and possible theoretical explanations are discussed in section 5, and conclusions are drawn in section 6.