# Improving Face Verification using Skin Color Information

Sebastien Marcel, Samy Bengio
Dalle Molle Institute for Perceptual Artificial Intelligence - IDIAP
Simplon 4
1920 Martigny, Switzerland
{marcel, bengio}@idiap.ch

## Abstract

*The performance of face verification systems has steadily improved over the last few years, mainly focusing on models rather than on feature processing. State-of-the-art methods often use the gray-scale face image as input. In this paper, we propose to use an additional feature to the face image: the skin color. The new feature set is tested on a benchmark database, namely XM2VTS, using a simple discriminant artificial neural network. Results show that the skin color information improves the performance.*

## 1. Introduction

Identity verification is a general task that has many real-life applications such as access control, transaction authentication (in telephone banking or remote credit card purchases for instance), voice mail, or secure teleworking.

The goal of an *automatic identity verification system* is to either accept or reject the identity claim made by a given person. Biometric identity verification systems are based on the characteristics of a person, such as its face, fingerprint or signature. A good introduction to identity verification can be found in [8]. Identity verification using face information is a challenging research area that was very active recently, mainly because of its natural and non-intrusive interaction with the authentication system.

In this paper, we investigate the use of skin color information as additional features in order to train face verification systems.

In the next section, we first introduce the reader to the problem of identity verification, based on face image (face verification). We present the model used and the proposed new feature set. We then compare this new set of features on the well-known benchmark database XM2VTS using its associated Lausanne protocol. Finally, we analyze the results and conclude.

## 2. Face Verification

An identity verification system has to deal with two kinds of events: either the person claiming a given identity is the one who he claims to be (in which case, he is called a *client*), or he is not (in which case, he is called an *impostor*). Moreover, the system may generally take two decisions: either *accept* the *client* or *reject* him and decide he is an *impostor*.

The classical face verification process can be decomposed into several steps, namely *image acquisition* (grab the images, from a camera or a VCR, in color or gray levels), *image processing* (apply filtering algorithms in order to enhance important features and to reduce the noise), *face detection* (detect and localize an eventual face in a given image) and finally *face verification* itself, which consists in verifying if the given face corresponds to the claimed identity of the client.

In this paper, we assume (as it is often done in comparable studies, but nonetheless incorrectly) that the detection step has been performed perfectly and we thus concentrate on the last step, namely the face verification step. A good survey on the different methods used in face verification can be found in [11].

## 3. The Proposed Approach

In face verification, we are interested in particular objects, namely faces. The representation used to code input images in most state-of-the-art methods are often based on gray-scale face image. In this section, we propose to use an additional feature to the face image: the skin color.

### 3.1. The Face Image as a Feature

In a real application, the face bounding box will be provided by an accurate face detector [7, 2], but here the bounding box is computed using manually located eyes coordinates, assuming a perfect face detection.

The face is cropped and the extracted sub-image is downsized to a 30x40 image. After enhancement and smoothing, the face image becomes a feature vector of dimension 1200. It is then possible to use this feature vector as the input of a face verification system (Figure 3). The objective of image enhancement is to modify the contrast of the image in order to enhance important features. On the other hand, smoothing is a simple algorithm which reduces the noise in the image (after image enhancement for example) by applying a Gaussian to the whole image.

### 3.2. The Skin Color as a Feature

Faces often have a characteristic color which is possible to separate from the rest of the image (Figure 1). Numerous methods exist to model the skin color, essentially using Gaussian mixtures [10] or simply using look-up tables.

In the present study, skin color pixels are filtered, from the sub-image corresponding to the extracted face, using a look-up table of skin color pixels. The skin color table was obtained by collecting, over a large number of color images, RGB (Red-Green-Blue) pixel values in sub-windows previously selected as containing only skin. The weak point of this method is the color similarity of hair pixels and skin pixels. For better results, the face bounding box should thus avoid as much hair as possible.

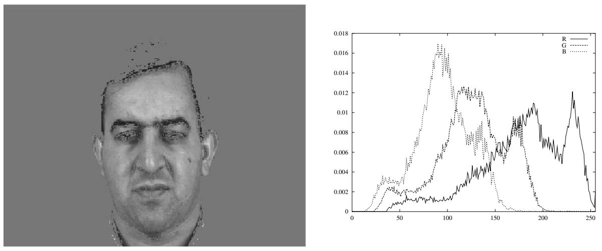

**Figure 1. Image and RGB distributions of filtered skin color pixels.**

As often done in skin color analysis studies [9], we compute the histogram of R, G and B pixel components for different face images. Such histograms are characteristic for a specific person, but are also discriminant among different persons (Figure 2).

Hence, we propose to use this characteristic information for a face verification system. In realistic situations, the use of normalised chrominance spaces (r-g) would yield more robust results. However, as a first valid attempt, the skin color feature for face verification is chosen to be simply the RGB color distribution of filtered pixels inside the face bounding box. For each color channel, an histogram is built using 32 discrete bins.

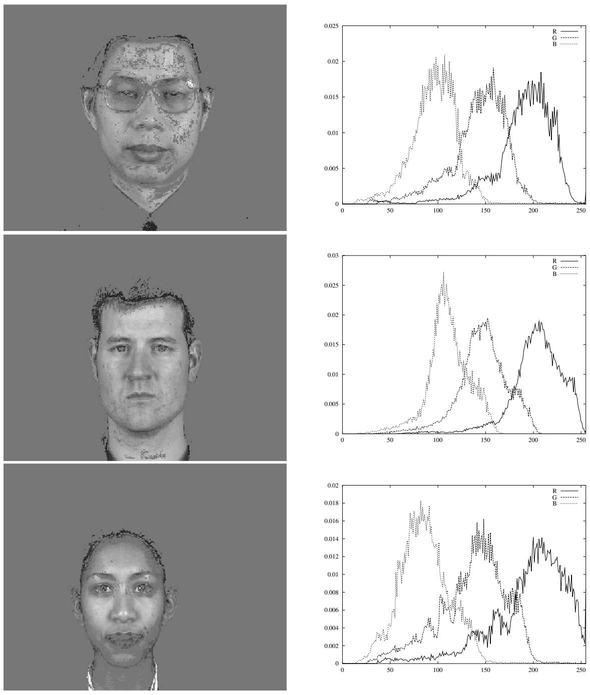

**Figure 2. Images and RGB distributions of skin color pixels of different persons (asian, european and african).**

The feature vector produced by the concatenation of the 3 histograms has 96 components (Figure 3).

### 3.3. The Model: a Discriminant Neural Network

The problem of face verification has been addressed by different researchers and with different methods. The aim of this section is not to propose a new model for face verification, but to present the model used to evaluate the new feature set.

Our face verification method is based on Multi-Layer Perceptrons (MLPs) [1]. For each client, an MLP is trained to classify an input to be either the given client or not. The input of the MLP is a feature vector corresponding to the face image with or without its skin color. The output of the MLP is either 1 (if the input corresponds to a client) or -1 (if the input corresponds to an impostor). The MLP is trained using both client images and impostor images, often taken to be the images corresponding to other available clients. In the present study, we used the other 199 clients of the XM2VTS database (see next section).

Finally, the decision to accept or reject a client access depends on the score obtained by the corresponding MLP which could be either above (accept) or under (reject) a given threshold, chosen on a separate validation set to optimize a given criterion.

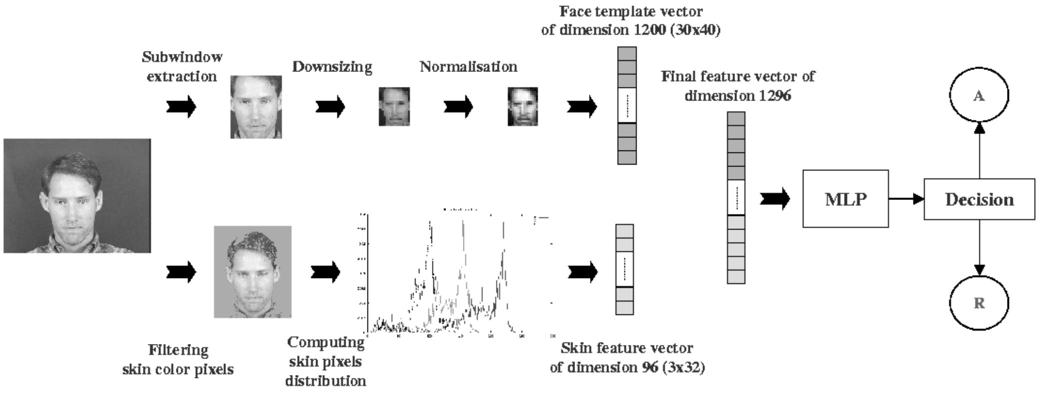


**Figure 3. An MLP for face verification using the image of the face and its skin color**

# 4. Experimental Results

In this section, we present an experimental comparison between two MLPs trained with and without skin color information. This comparison has been done using the multimodal XM2VTS database, using its associated experimental protocol, the *Lausanne Protocol* [5].

## 4.1. The Database and the Protocol

The XM2VTS database contains synchronized image and speech data recorded on 295 subjects during four sessions taken at one month intervals. On each session, two recordings were made, each consisting of a speech shot and a head rotation shot.

The database was divided into three sets: a training set, an evaluation set, and a test set. The training set was used to build client models, while the evaluation set was used to compute the decision (by estimating thresholds for instance, or parameters of a fusion algorithm). Finally, the test set was used only to estimate the performance of the two different features.

The 295 subjects were divided into a set of 200 clients, 25 evaluation impostors, and 70 test impostors. Two different evaluation configurations were defined. They differ in the distribution of client training and client evaluation data. Both the training client and evaluation client data were drawn from the same recording sessions for Configuration I which might lead to biased estimation on the evaluation set and hence poor performance on the test set. For Configuration II on the other hand, the evaluation client and test client sets are drawn from different recording sessions which might lead to more realistic results. Hence, we have decided to perform our experiments using Configuration II only. For each client, there is $4$ training shot.

The system may make two types of errors: *false acceptances* (FA), when the system accepts an *impostor*, and *false rejections* (FR), when the system rejects a *client*. In order to be independent on the specific dataset distribution, the performance of the system is often measured in terms of these two different errors, as follows:

$$\text{FAR} = \frac{\text{number of FAs}}{\text{number of impostor accesses}}\,, \tag{1}$$

$$\text{FRR} = \frac{\text{number of FRs}}{\text{number of client accesses}}\,. \tag{2}$$

A unique measure often used combines these two ratios into the so-called *Half Total Error Rate* (HTER) as follows:

$$\text{HTER} = \frac{\text{FAR} + \text{FRR}}{2}\,. \tag{3}$$

Most verification systems output a score for each access. Selecting a threshold over which scores are considered genuine clients instead of impostors can greatly modify the relative performance of FAR and FRR. A typical threshold chosen is the one that reaches the *Equal Error Rate* (EER) where FAR=FRR on a separate validation set.

## 4.2. Comparative Results

We have compared an MLP using $1200$ inputs corresponding to the downsized (30x40) gray-scale face image and an MLP using $1296$ inputs corresponding to the same face image as well as its skin color distribution. Configuration II of the **Lausanne Protocol** is chosen for these comparative experiments as it is the most realistic configuration.

For each client model, the training database is composed of a client training set and an impostor training set. As often done in comparable studies, the client training set is enlarged by shifting (8 directions and 4 pixel shifts), scaling (2 scales) and mirroring the original face bounding box. Hence, the client training set contains $1320$ patterns ($4 * P$) instead of $4$.

The extended number of pattern $P$ is computed such that $P = 2 * A * B$, i.e. the mirrored number of shifted and scaled face patterns. $A =$ number of shifts $* 8 + 1$ is the total number of shifts, in 8 directions, including the original frame, for each scale. $B =$ number of scales $* 2 + 1$ is the total number of scales, in 2 directions (sub-scaling and over-scaling), including the original scale.

On the other hand, the impostor training set contains 796 patterns (the 4 original patterns of each of the 199 other clients). These training sets are then divided into three subsets: a training set, a validation set and a test set. The training set is used to train the MLP, the validation set is used to stop the training using an early-stopping criterion and the test set is used to choose the best MLP architecture. The chosen architecture is an MLP with 90 hidden units.

The trained model is used on the evaluation set to evaluate the global threshold that optimized the EER. This threshold is then used with the same trained model on the test set to compute the HTER. Results are shown in Table 1. This table provides the FAR, FRR and HTER on the test set, both for the MLP using only the 30x40 face image and for the MLP using the 30x40 face image and its 96 skin color vector. These results show a good improvement when using the skin color information.

| Features | FAR | FRR | HTER |
|---|---|---|---|
| Without skin color | 2.364 | 3.250 | 2.807 |
| With skin color | 1.499 | 2.750 | **2.125** |

**Table 1. Comparative results with and without the use of the skin color**

These results are competitive when compared to recent results published on the same database. In [6] for instance, the best face HTER (with global thresholds) was 1.5 on the same data using LDA [4] and 61x57 face images from all the XM2VTS training set, i.e more training data and images 3 times bigger than proposed in this paper.

## 5. Conclusion

In this paper, we have proposed to use the skin color information in addition to the face image to improve face verification systems. Experimental comparisons have been carried out using the XM2VTS benchmark database. Results have shown that the skin color distribution of the face increases the performance. These results can be improved further using all the XM2VTS training set. For example, additional experiments with the same MLP architecture and the full training set give an HTER as low as 1.73 using skin color information.

More recently, using a special combination algorithm, ECOC [3], normally designed for robust multi-class classification tasks, researchers were able to obtain an HTER as low as 0.80 on the face verification task using configuration I of XM2VTS and only a 28x28 face image, but no comparable results were published for configuration II. The use of such a model with the feature proposed in this paper should probably lead to further performance improvements.

**Acknowledgments** This research has been carried out in the framework of the European BANCA project, funded by the Swiss OFES project number 99-0563-1.

## References

[1] C. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

[2] R. Féraud, O. Bernier, J.-E. Viallet, and M. Collobert. A fast and accurate face detector based on neural networks. *Transactions on Pattern Analysis and Machine Intelligence*, 23(1), 2001.

[3] J. K. J, R. Ghaderi, T. Windeatt, and G. Matas. Face verification via ECOC. In *British Machine Vision Conference (BMVC01)*, pages 593–602, 2001.

[4] Y. Li, J. Kittler, and J. Matas. On matching scores of LDA-based face verification. In T. Pridmore and D. Elliman, editors, *Proceedings of the British Machine Vision Conference BMVC2000*. British Machine Vision Association, 2000.

[5] J. Lüttin and G. Maître. Evaluation protocol for the extended M2VTS database (XM2VTSDB). Technical Report RR-21, IDIAP, 1998.

[6] J. Matas, M. Hamouz, K. Jonsson, J. Kittler, Y. Li, C. Kotropoulos, A. Tefas, I. Pitas, T. Tan, H. Yan, F. Smeraldi, J. Bigun, N. Capdevielle, W. Gerstner, S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz. Comparison of face verification results on the XM2VTS database. In A. Sanfeliu, J. J. Villanueva, M. Vanrell, R. Alqueraz, J. Crowley, and Y. Shirai, editors, *Proceedings of the 15th ICPR*, volume 4, pages 858–863. IEEE Computer Society Press, 2000.

[7] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *Transactions on Pattern Analysis and Machine Intelligence*, 20(1), 1998.

[8] P. Verlinde, G. Chollet, and M. Acheroy. Multi-modal identity verification using expert fusion. *Information Fusion*, 1:17–33, 2000.

[9] J. Yang, W. Lu, and A. Waibel. Skin color modeling and adaptation. In *Proceedings of the $3^{rd}$ Asian Conference on Computer Vision*, volume 2, pages 687–694, 1998.

[10] M.-H. Yang and N. Ahuja. Gaussian Mixture Model for Human Skin Color and Its Applications in Image and Video Databases. In *Conference on Storage and Retrieval for Image and Video Databases*, volume 3656, pages 458–466, 1999.

[11] J. Zhang, Y. Yan, and M. Lades. Face recognition: Eigenfaces, elastic matching, and neural nets. In *Proceedings of IEEE*, volume 85, pages 1422–1435, 1997.