# A STATE-OF-THE-ART NEURAL NETWORK FOR ROBUST FACE VERIFICATION

*Christine Marcel* *Sébastien Marcel* *Samy Bengio*

Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP)
P. O. Box 592, CH-1920 Martigny, Switzerland
{alves,marcel}@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 and that the proposed model achieves robust state-of-the-art results.

## 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 [17]. 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 using artificial neural networks. 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

### 2.1. Problem Description

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.

### 2.2. State-of-the-art methods

The problem of face verification has been addressed by different researchers and with different methods. For a complete survey and comparison of different approaches see [4].In this section, we briefly introduce one of the best method [10]. This method adopts a client-specific solution which requires learning client-specific support vectors. Faces are represented in both Principal Component and Linear Discriminant subspaces.

The aim of the Principal Component Analysis (PCA) is to identify the subspace of the image space spanned by the training face image data and to decorrelate the pixel values. This can be achieved by finding the eigenvectors of matrix associated with nonzero eigenvalues. These eigenvectors are referred to as Eigenfaces. The classical representation of a face image is obtained by projecting it to the coordinate system defined by the Eigenfaces. The projection of face images into the Principal Component (Eigenface) subspace achieves information compression, decorrelation and dimensionality reduction to facilitate decision making. If one is also interested in identifying important attributes (features) for face verification, one can adopt a feature extraction mapping. A popular technique is to find the Fisher linear discriminant [6].

The linear discriminant analysis (LDA) subspace holds more discriminant features for classification than the PCA subspace. The LDA based features for personal identity verification is theorically superior to that achievable with the features computed using PCA [16] and many others [1, 5]. The projection of a face image into the system of Fisher-faces associated with nonzero eigenvalues will yield a representation which will emphasize the discriminatory content of the image. The main decision making tool is Support Vector Machines (SVMs). The reader is referred to [3] for a comprehensive introduction of SVMs.

## 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 [15, 7], 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 $30$x$40$ 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 (Fig. 1). 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

Skin color has already been used successfully for face detection [7] but, to our knowledge, not to face verification. Faces often have a characteristic color which is possible to separate from the rest of the image. Numerous methods exist to model the skin color, essentially using Gaussian mixtures [19] 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.

As often done in skin color analysis studies [18], 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 [13].

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. Furthermore, images used in this study were recorded in controlled environment (blue background) with constant lighting conditions. Thus, we are not facing the problem of color identification under changes in illumination.

For each color channel, an histogram is built using $32$ discrete bins. Hence, the feature vector produced by the concatenation of the $3$ histograms (R, G and B) has $96$ components (Fig. 1).

### 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) [2, 8]. 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.

## 4. EXPERIMENTAL RESULTS

In this section, we present an experimental[1] comparison between two MLPs trained with and without skin color information. This comparison has been done using the multi-modal XM2VTS database and its associated experimental protocol, the **Lausanne Protocol** (LP) [12].

### 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 (LP1) which might lead to biased estimation on the evaluation set and hence poor performance on the test set. For Configuration II (LP2) on the other hand, the evaluation client and test client sets are drawn from different recording sessions which might lead to more realistic results. This led to the following statistics:

- Training client accesses: $3$ for LP1 and $4$ for LP2
- Evaluation client accesses: $600$ for LP1 and $400$ for LP2
- Evaluation impostor accesses: $40,000$ $(25 * 8 * 200)$
- Test client accesses: $400$ $(200 * 2)$
- Test impostor accesses: $112,000$ $(70 * 8 * 200)$

Thus, 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:

$$\text{FAR} = \frac{\text{number of FAs}}{\text{number of impostor accesses}} , \quad (1)$$

$$\text{FRR} = \frac{\text{number of FRs}}{\text{number of client accesses}} . \quad (2)$$

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

$$\text{HTER} = \frac{\text{FAR} + \text{FRR}}{2} . \quad (3)$$

[1]The machine learning library used for all experiments is Torch http://www.torch.ch.

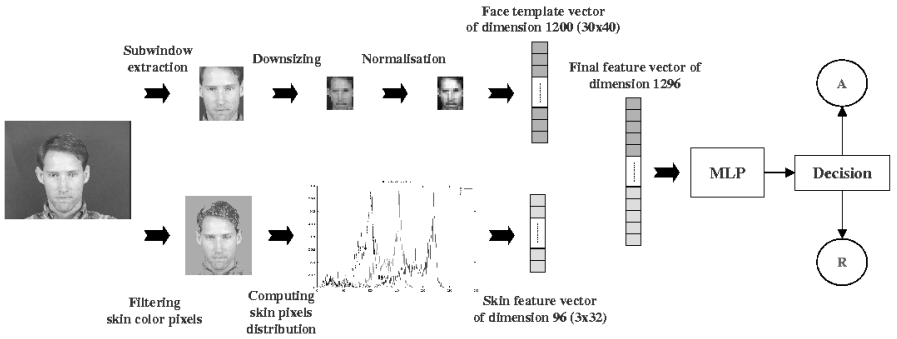


**Fig. 1**. An MLP for face verification using the image of the face and its skin color

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. Experiment 1: Improving Results using Skin Color

We have compared an MLP using $1200$ inputs corresponding to the downsized ($30$x$40$) gray-scale face image and an MLP using $1296$ inputs corresponding to the same face image as well as its skin color distribution [13]. 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 (4 images) 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 sub-sets: 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 LP evaluation set to evaluate the global threshold that optimized the EER. This threshold is then used with the same trained model on the LP 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 These results show a good improvement when using the skin color information.

### 4.3. Experiment 2: Comparison to State-of-the-art

We have trained our best MLP architecture (face image and skin color) using all the XM2VTS training set on both configurations.

| 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

For each client model, the training database is composed of a client training set (3 images for LP1 and 4 images for LP2) and an impostor training set. Again, 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 $990$ patterns ($3 * P$) for LP1 and $1320$ patterns ($4 * P$) for LP2. On the other hand, the impostor training set contains $1194$ patterns for LP1 and $1592$ patterns for LP2 (the mirrored 4 original patterns of each of the 199 other clients).

These training sets are not divided into sub-sets. All training sets are used to train the 200 MLPs (one for each client). The chosen architecture is the one selected during experiment 1: an MLP with $90$ hidden units. Furthermore, the training is stopped when the number of iterations is equal to the number of iterations obtained when the learning process converged during experiment 1.

Then, as previously described, the global threshold optimizing the EER is evaluated on the LP evaluation set and the corresponding HTER is computed on the LP test set. This leads to an HTER lower than **1.9** on both configurations (Fig. 2 and 3).

| Model | FAR | FRR | HTER |
|---|---|---|---|
| NC | 3.46 | 2.75 | 3.1 |
| MLP | 1.75 | 2.00 | **1.87** |

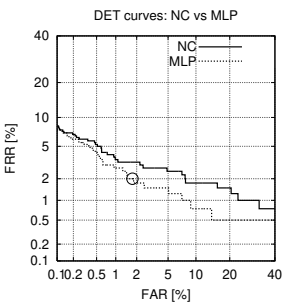


**Fig. 2**. Comparative results (left) and DET curves (right) on the configuration 1 of NC and the proposed MLP using the image of the face and its skin color.

These results are competitive when compared to recent results

published on the same database and the same protocol. In [14] for instance, the best face HTER (with global thresholds) was obtained using Normalized Correlation (NC) [11] and 61x57 face images from all the XM2VTS training set, i.e images 3 times bigger than proposed in this paper. Our MLP yields better results than NC on LP1 and slightly worse results on LP2. However, the proposed model is robust over both configurations and achieves state-of-the-art average results: 1.86 HTER for the MLP versus 2.3 HTER for NC.

| Model | FAR | FRR | HTER |
|---|---|---|---|
| NC | 1.26 | 1.75 | **1.50** |
| MLP | 1.46 | 2.25 | 1.85 |

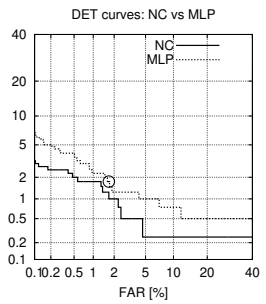


**Fig. 3**. Comparative results (left) and DET curves (right) on the configuration 2 of NC and the proposed MLP using the image of the face and its skin color.

## 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. Results have shown also that the proposed model is robust in all configurations and achieves state-of-the-art results.

More recently, using a special combination algorithm, ECOC [9], 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.

## 6. ACKNOWLEDGMENTS

This work was funded by the European project "BANCA", through the Swiss Federal Office for Education and Science (OFES).

## 7. REFERENCES

[1] P. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," in **ECCV'96**, 1996, pp. 45–58, Cambridge, United Kingdom.

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

[3] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," **Data Mining and Knowledge Discovery**, vol. 2, no. 2, pp. 1–47, 1998.

[4] R. Chellappa, C. Wilson and C. Barnes, "Human and machine recognition of faces: A survey," Techn. Report CAR-TR-731, **University of Maryland**, 1994.

[5] P. A. Devijver and J. Kittler, **Pattern Recognition: A Statistical Approach**, Prentice-Hall, Englewood Cliffs, N.J., 1982.

[6] R. A. Fisher, "The use of multiple measurements in taxonomic problems," **Annals of Eugenics**, vol. 7, no. II, pp. 179–188, 1936.

[7] R. Fraud, 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**, vol. 23, no. 1, 2001.

[8] S. Haykin, **Neural Networks, a Comprehensive Foundation, second edition**, Prentice Hall, 1999.

[9] J. K. J, R. Ghaderi, T. Windeatt and G. Matas, "Face Verification via ECOC," in **British Machine Vision Conference (BMVC01)**, 2001, pp. 593–602.

[10] K. Jonsson, J. Matas, J. Kittler and Y. Li, "Learning Support Vectors for Face Verification and Recognition," in 4**th International Conference on Automatic Face and Gesture Recognition**, 2000, pp. 208–213.

[11] Y. Li, J. Kittler and J. Matas, "On Matching Scores of LDA-based Face Verification," in T. Pridmore and D. Elliman (eds.), **Proceedings of the British Machine Vision Conference BMVC2000**, British Machine Vision Association, 2000.

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

[13] S. Marcel and S. Bengio, "Improving Face Verification using Skin Color Information," in **Proceedings of the 16th ICPR**, IEEE Computer Society Press, 2002.

[14] 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 (eds.), **Proceedings of the 15th ICPR**, IEEE Computer Society Press, 2000, vol. 4, pp. 858–863.

[15] H. A. Rowley, S. Baluja and T. Kanade, "Neural Network-Based Face Detection," **Transactions on Pattern Analysis and Machine Intelligence**, vol. 20, no. 1, 1998.

[16] M. Turk and A. Pentland, "Eigenface for Recognition," **Journal of Cognitive Neuro-science**, vol. 3, no. 1, pp. 70–86, 1991.

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

[18] J. Yang, W. Lu and A. Waibel, "Skin color modeling and adaptation," in **Proceedings of the 3**$^{rd}$ **Asian Conference on Computer Vision**, 1998, vol. 2, pp. 687–694.

[19] 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**, 1999, vol. 3656, pp. 458–466.