Discriminant Machine Learning Algorithms for Face Verification: MLP and SVM, an Experimental Comparison

Sébastien Marcel *, Fabien Cardinaux

Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP)
CP 592, rue du Simplon 4, 1920 Martigny, Switzerland

Abstract

The performance of face verification systems has steadily improved over the last few years. State-of-the-art methods are often based on discriminant machine learning algorithms and often use the gray-scale face image as input. In this paper, we propose an experimental comparison of two famous machine learning algorithms: Multi-Layer Perceptrons (MLPs) and Support Vector Machines (SVMs). We propose also 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 MLP and a SVM. Results show that the skin color information improves the performance, that the MLP outperform the SVM on this particular task and that the same MLP achieves robust state-of-the-art results.

Key words: face verification, machine learning algorithms, MLP, SVM

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

* Corresponding author.
Email address: marcel@idiap.ch (Sébastien Marcel).

in [1]. 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 use the skin color information as an additional feature to the face image in order to train face verification systems using MLPs and SVMs. 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 [2,3]. In this section, we briefly introduce one of the best method [4]. 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 [5].

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 theoretically superior to that achievable with the features computed using PCA [6] and many others [7,8]. 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).

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 [9,10], 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 (Fig. 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

Skin color has already been used successfully in face detection [10] but, to our knowledge, not in face verification. Still, skin color provide an additional information that improve results [11]. Faces often have a characteristic color which is possible to separate from the rest of the image (Fig. 1). Numerous methods exist to model the skin color, essentially using Gaussian mixtures [12] or simply using look-up tables [13].

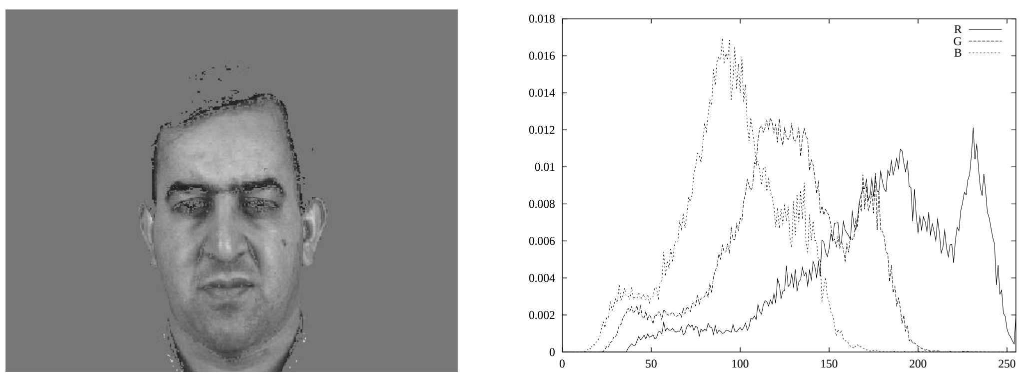

Fig. 1. Image and RGB distributions of filtered skin color pixels.

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 [14], we compute the histogram of R, G and B pixel components for different face images. Such histograms are characteristic (Fig. 1) for a specific person, but are also discriminant among different persons [15] (Fig. 2).

Hence, we propose to use this characteristic information for a face verification system. In realistic situations, the use of normalized 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

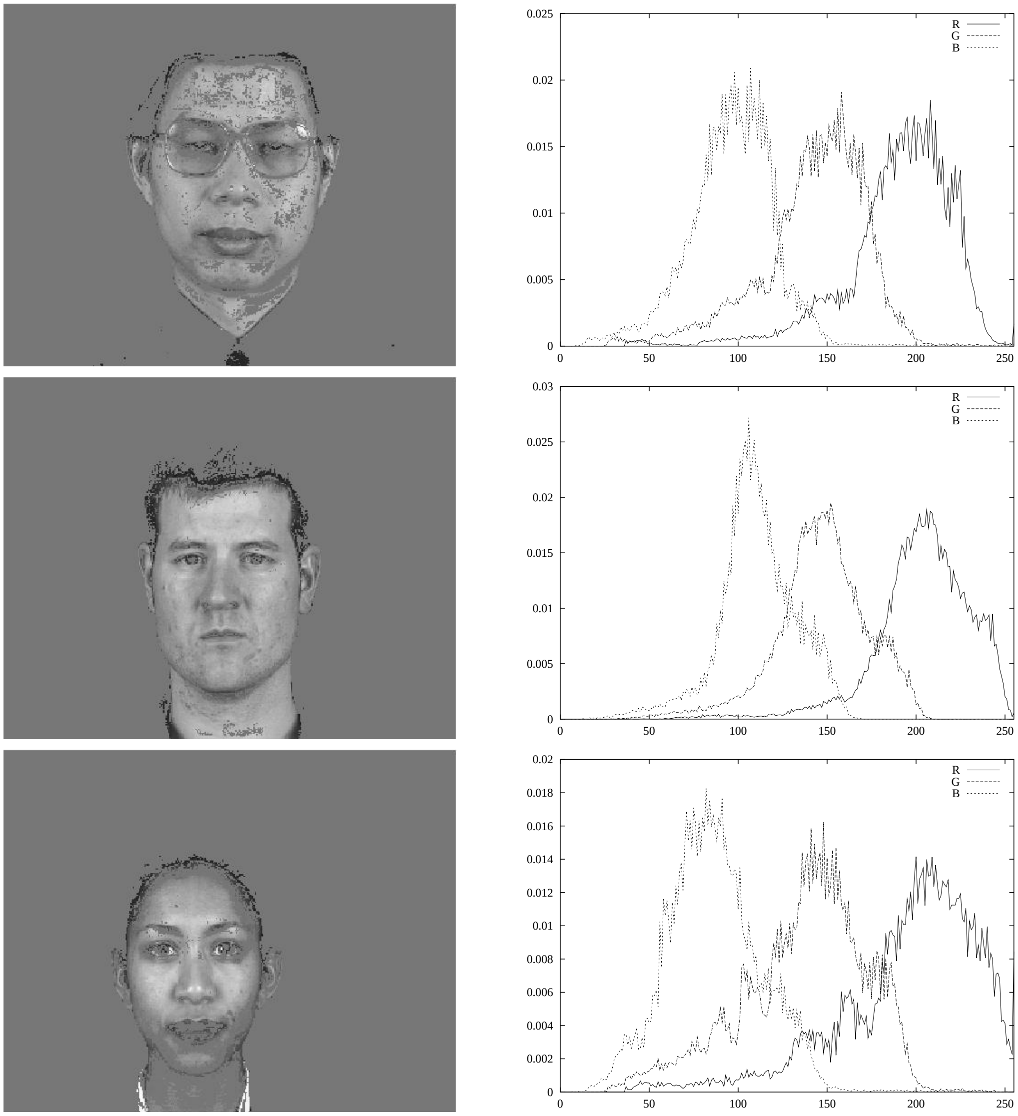

Fig. 2. Images and RGB distributions of skin color pixels of different persons (Asian, European and African).

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. 3).

We propose to use and to compare two machine learning algorithms for face verification: Multi-Layer Perceptrons (MLPs) and Support Vector Machines (SVMs). MLPs and SVMs have been used already for face verification [11][16][4]. The aim of this paper is not to introduce these algorithms as new approaches for face verification but to provide an experimental comparison of these two famous machine learning algorithms with the same input features and the same database.

For each client, a model (MLP or SVM) is trained to classify an input to be either the given client or not. The input of the model is a feature vector corresponding to the face image. The output of the model is either 1 (if the input corresponds to the client) or -1 (if the input corresponds to an impostor). The model 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 model which could be either above (accept) or under (reject) a given threshold, chosen on a separate validation set to optimize a given criterion.

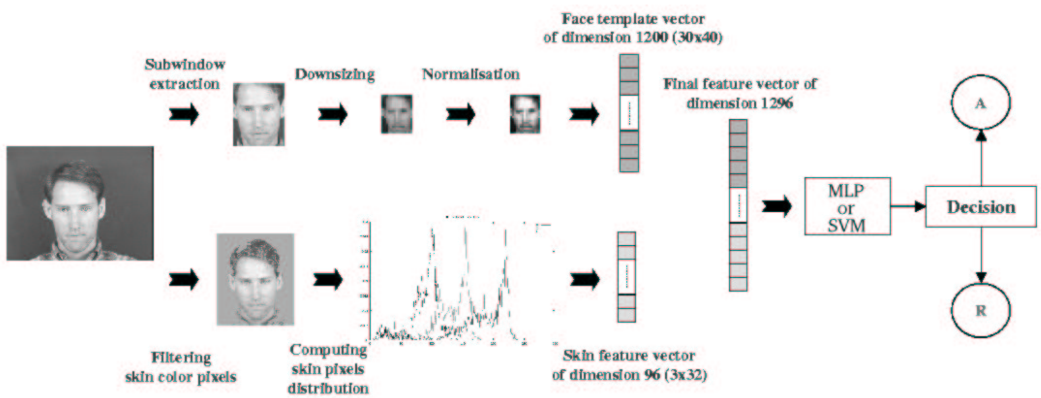

Fig. 3. Face verification using the image of the face and its skin color

3.4 Introduction to MLPs and SVMs

MLPs and SVMs are learning machines used in many classification problems. A good introduction to machine learning algorithms can be found in [17,18]. We will assume that we have access to a training dataset of l pairs $(\mathbf{x}_i, y_i)$ where $\mathbf{x}_i$ is a vector containing the pattern, while y_i is the class of the corresponding pattern often coded respectively as 1 and -1.

3.4.1 Multi-Layer Perceptron

A MLP is a particular architecture of artificial neural networks [17,18], composed of layers of non-linear but differentiable parametric functions. For instance, the output $\hat{y}$ of a 1-hidden-layer MLP can be written mathematically as follows

$$\hat{y} = b + \mathbf{w} \cdot \tanh\left(\mathbf{a} + \mathbf{x} \cdot \mathbf{V}\right) \tag{1}$$

where the estimated output $\hat{y}$ is a function of the input vector $\mathbf{x}$, and the parameters $\{b, \mathbf{w}, \mathbf{a}, \mathbf{V}\}$. In this notation, the non-linear function tanh() returns a vector which size is equal to the number of hidden units of the MLP, which controls its capacity and should thus be chosen carefully, by cross-validation for instance.

An MLP can be trained by gradient descent using the backpropagation algorithm [19] to optimize any derivable criterion, such as the *mean squared error* (MSE):

$$\text{MSE} = \frac{1}{l} \sum_{i=1}^{l} (y_i - \hat{y}_i)^2 \, . \tag{2}$$

3.4.2 Support Vector Machine

SVMs have been introduced by V. Vapnik [20].The SVM algorithm constructs a separating hypersurface in the input space. It acts as follows:

- maps the input space into a higher dimensional feature space through some nonlinear mapping chosen *a priori* (kernel);
- constructs the maximal margin hyperplane in this feature space (MMH); the MMH maximizes the distance of the closest vectors belonging to the different classes to the hyperplane.

The resulting function is of the form:

$$\hat{y} = \text{sign}\left(\sum_{i=1}^{l} y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b\right) \tag{3}$$

where $\mathbf{x}$ is the input vector of a example to test, $\hat{y}$ is the decision of the model (accept if $\hat{y}$ is positive, reject otherwise), $\mathbf{x}_i$ is the input vector of the i^{th} training example, l is the number of training examples, the α_i and b are the parameters of the model, and $K(\mathbf{x}, \mathbf{x}_i)$ is a kernel function that can have different forms, such as:

$$K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x}^t \, \mathbf{x}_i + 1)^d \tag{4}$$

which leads to a Polynomial SVM with parameter d, or

$$K(\mathbf{x}, \mathbf{x}_i) = \exp\left(\frac{-\|\mathbf{x} - \mathbf{x}_i\|^2}{\sigma^2}\right) \tag{5}$$

which leads to a Radial Basis Function (RBF) SVM with parameter σ. Either d or σ must be selected using methods such as cross-validation.

In order to train such SVMs, one needs to solve the following quadratic optimization problem: find the parameter vector $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \ldots, \alpha_l\}$ that maximize the objective function

$$Q(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{6}$$

subject to the constraints:

$\sum_{i=1}^{l} \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C \; \forall i$.

It is important to note that the training complexity of SVMs is quadratic on the number l of examples, which makes the use of SVMs for large datasets difficult. Note however that in the resulting solution (3), most α_i are equal to 0, and the examples with non-zero α_i are called *support vectors*.

The reader is referred to [21] for a comprehensive introduction of SVMs.

4 Experimental Results

In this section, we present an experimental [1] comparison between two MLPs trained with and without skin color information and also experimental comparisons between MLPs and SVMs. These comparisons have been done using the multi-modal XM2VTS database and its associated experimental protocol, the **Lausanne Protocol** (LP) [22].

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

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

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:

$$\text{FAR} = \frac{\text{number of FAs}}{\text{number of impostor accesses}} , \tag{7}$$

$$\text{FRR} = \frac{\text{number of FRs}}{\text{number of client accesses}} . \tag{8}$$

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

$$\text{HTER} = \frac{\text{FAR} + \text{FRR}}{2} . \tag{9}$$

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 a MLP using 1200 inputs corresponding to the downsized (30x40) gray-scale face image and a MLP using 1296 inputs corresponding to the same face image as well as its skin color distribution [15]. 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 a MLP with 90 hidden units.

Features	FAR	FRR	HTER
Face image without skin color	2.364	3.250	2.807
Face image with skin color	1.499	2.750	**2.125**

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

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.

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

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: a 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. 4 and 5).

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

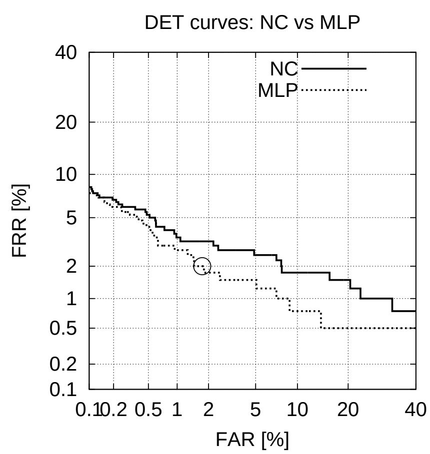

Fig. 4. Comparative results (left) and DET curves (right) on the configuration 1 of NC and the proposed MLP using the image of the face and its skin color.

These results are competitive when compared to recent results published on the same database and the same protocol. In [23] for instance, the best face HTER (with global thresholds) was obtained using Normalized Correlation (NC) [24] and 61x57 face images from all the XM2VTS training set, i.e images

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

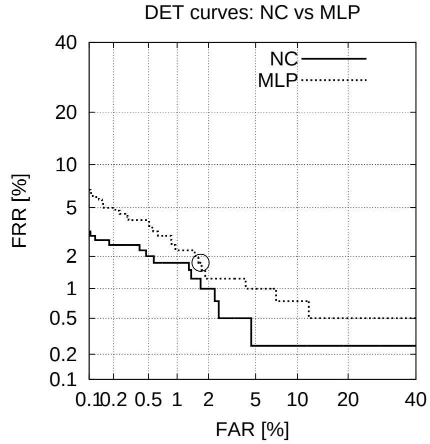

Fig. 5. Comparative results (left) and DET curves (right) on the configuration 2 of NC and the proposed MLP using the image of the face and its skin color.

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.

4.4 Experiment 3: Comparison to SVMs

We have compared also the face verification models based on MLPs to face verification models based on SVMs. The Configuration I of the **Lausanne Protocol** is chosen for these experiments.

The same training database than for MLPs (Experiment 2) is used to train the 200 SVMs (one for each client). The chosen kernel for SVMs is a Gaussian kernel ($\sigma = 11$) while the chosen architecture for MLPs had 1 hidden layer with 90 hidden units.

The trained model is used on the evaluation set to evaluate the global threshold that optimized the EER (Table 2). This threshold is then used with the same trained model on the test set to compute the HTER (Table 3).

Model	FAR	FRR	HTER
MLP	1.67	1.67	**1.67**
SVM	1.70	1.67	**1.69**

Table 2
Comparative results of MLPs and SVMs on the Evaluation set

Each table provides the FAR, FRR and HTER, both for MLPs and for SVMs. Results show that face verification using MLPs gives better results than using SVMs.

Model	FAR	FRR	HTER
MLP	1.75	2.00	**1.87**
SVM	1.84	3.25	2.54

Table 3
Comparative results of MLPs and SVMs on the Test set

5 Conclusion

MLPs and SVMs have been applied to many classification problems, generally yielding good performance compared to other algorithms. In this paper, we have compared these two machine learning algorithms on face verification. We have also used 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. First, results have shown that the skin color distribution of the face increases the performance on this particular database. This conclusion is true only under controlled environments and constant lighting conditions. Thus, it is highly probable that the skin color information, as used in this paper, will not improve the performance when using degraded conditions. Anyway, depending on the lighting conditions the skin color feature could be turned off without increasing drastically the performance (Table 1). Results have shown also that a MLP outperform a SVM on this particular task. Indeed, HTER obtained using MLPs and SVMs for the evaluation set are equivalent. However on the test set, the HTER obtained by MLPs is quite better than HTER obtained by SVMs. This experimental comparison show that MLPs have a better generalization performance than SVMs in this face verification tasks. Furthermore, the MLP is robust in all configurations and achieves state-of-the-art results.

More recently, using a special combination algorithm, ECOC [25], 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 The authors wish to thank S. Bengio for fruitful discussions and collaboration. The author want to thank the Swiss National Science Foundation for supporting this work through the National Center of Competence in Research (NCCR) on "Interactive Multimodal Information Management (IM2)". This work was also funded by the European project "CIMWOS", through the Swiss Federal Office for Education and Science (OFES).

References

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

[2] R. Chellappa, C. Wilson, C. Barnes, Human and machine recognition of faces: A survey, Tech. Rep. CAR-TR-731, University of Maryland (1994).

[3] J. Zhang, Y. Yan, M. Lades, Face recognition: Eigenfaces, elastic matching, and neural nets, in: Proceedings of IEEE, Vol. 85, 1997, pp. 1422–1435.

[4] K. Jonsson, J. Matas, J. Kittler, Y. Li, Learning support vectors for face verification and recognition, in: 4th International Conference on Automatic Face and Gesture Recognition, 2000, pp. 208–213.

[5] R. A. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 7 (II) (1936) 179–188.

[6] M. Turk, A. Pentland, Eigenface for recognition, Journal of Cognitive Neuroscience 3 (1) (1991) 70–86.

[7] P. Belhumeur, J. P. Hespanha, D. J. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, in: ECCV'96, 1996, pp. 45–58, cambridge, United Kingdom.

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

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

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

[11] S. Marcel, S. Bengio, Improving face verification using skin color information, in: 16th International Conference on Pattern Recognition, 2002.

[12] M.-H. Yang, 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, Vol. 3656, 1999, pp. 458–466.

[13] R. Feraud, O. Bernier, M. Collobert, A fast and accurate face detector for indexation of face images, in: 4th IEEE International Conference on Automatic Face and Gesture Recognition, 2000.

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

[15] S. Marcel, S. Bengio, Improving face verification using skin color information, in: Proceedings of the 16th ICPR (to appear), IEEE Computer Society Press, 2002.

[16] K. Jonsson, J. Kittler, Y. Li, J. Matas, Support vector machines for face authentication, in: British Machine Vision Conference (BMVC99), 1999, pp. 543–553.

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

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

[19] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning internal representations by error propagation, in: D. E. Rumelhart, J. L. McClelland (Eds.), Parallel Distributed Processing, Vol. 1, MIT Press, Cambridge, MA., 1986.

[20] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.

[21] C. J. C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2 (2) (1998) 1–47.

[22] J. Lüttin, G. Maître, Evaluation protocol for the extended M2VTS database (XM2VTSDB), Tech. Rep. RR-21, IDIAP (1998).

[23] 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, E. Mayoraz, Comparison of face verification results on the XM2VTS database, in: A. Sanfeliu, J. J. Villanueva, M. Vanrell, R. Alqueraz, J. Crowley, Y. Shirai (Eds.), Proceedings of the 15th ICPR, Vol. 4, IEEE Computer Society Press, 2000, pp. 858–863.

[24] Y. Li, J. Kittler, J. Matas, On matching scores of LDA-based face verification, in: T. Pridmore, D. Elliman (Eds.), Proceedings of the British Machine Vision Conference BMVC2000, British Machine Vision Association, 2000.

[25] J. K. J, R. Ghaderi, T. Windeatt, G. Matas, Face verification via ECOC, in: British Machine Vision Conference (BMVC01), 2001, pp. 593–602.