# FACE DETECTION USING DISCRETE GABOR JETS AND COLOR INFORMATION

Ulrich Hoffmann[1], Jacek Naruniec[2], Ashkan Yazdani[1] and Touradj Ebrahimi[1]
[1] *Multimedia Signal Processing Group, Ecole Polytechnique Fédérale de Lausanne (EPFL)*
*CH-1015, Lausanne, Switzerland*
[2] *Faculty of Electronics and Information Technology,Warsaw University of Technology, 00-665 Warszawa, Poland*
*ulrich.hoffmann@gmail.com, j.naruniec@ire.pw.edu.pl, {ashkan.yazdani, touradj.ebrahimi}@epfl.ch*

Keywords: Face Detection, Colored Image Patch Model, Discrete Gabor Jets, Linear Discriminant Analysis.

Abstract: Face detection allows to recognize and detect human faces and provides information about their location in a given image. Many applications such as biometrics, face recognition, and video surveillance employ face detection as one of their main modules. Therefore, improvement in the performance of existing face detection systems and new achievements in this field of research are of significant importance. In this paper a hierarchical classification approach for face detection is presented. In the first step, discrete Gabor jets (DGJ) are used for extracting features related to the brightness information of images and a preliminary classification is made. Afterwards, a skin detection algorithm, based on modeling of colored image patches, is employed as a post-processing of the results of DGJ-based classification. It is shown that the use of color efficiently reduces the number of false positives while maintaining a high true positive rate. Finally, a comparison is made with the OpenCV implementation of the Viola and Jones face detector and it is concluded that higher correct classification rates can be attained using the proposed face detector.

## 1 INTRODUCTION

The goal of face detection is to automatically find faces in digital images. Given the pixels of an input image, a face detection algorithm should return the number of faces in that image and their coordinates. The motivation for studying such algorithms is that face detection is an important module in many applications involving digital images or video. One example for an application area where face detection plays an important role is biometric authentication based on face recognition. Other examples of applications involving face detection are automatic lip reading, facial expression recognition, advanced teleconferencing, video surveillance, and automatic adjustment of exposure and focus in modern digital cameras.

Given the many possible applications, it is no wonder that many different methods already exist for face detection (see (Yang et al., 2001) for a survey). One of the most well-known methods for detecting upright frontal faces is based on a cascade of classifiers trained using Adaboost and features resembling Haar-wavelet bases (Viola and Jones, 2001). Other well-known methods are the neural network based method presented in (Rowley et al., 1998) and the method presented in (Sung and Poggio, 1998). A unifying feature of the methods described above and of many other current approaches is that they are based solely on features computed from the brightness of pixels. This means faces are detected by analyzing brightness patterns in rectangular image patches.

An alternative to using only brightness patterns is to employ also features related to the color of pixels. The rationale underlying such an approach is that many objects can be distinguished from other objects based on their color. For face detection skin color is an important cue, indicating that an image patch might contain a face. The main advantages of using color information for object detection tasks are that it is robust against rotations, changes in scale, and partial occlusions.

An early example for a face detection system using skin color is the system described in (Yang and Ahuja, 1998). In this system skin pixels are detected with a probabilistic model and skin regions are segmented with a multiscale segmentation. Skin regions having an elliptical shape and other facial characteristics are then classified as faces. A similar method was presented in (Hsu et al., 2002). In this work, first illumination compensation is performed, then color features are used to detect skin regions, eyes, and mouth, and finally eye-mouth triangles are computed to detect face candidates. Different from the work presented in (Yang and Ahuja, 1998; Hsu et al., 2002), the method in (Feraud et al., 2001), uses color information for prefiltering images. This means regions without skin color are rejected before further processing of an image takes place. In the algorithm described in (Huang et al., 2004), color is integrated in the face detection process by transforming images into YCrCb space, extracting wavelet features from each color channel, and finally by combining the wavelet features with the help of Bayesian classifiers and Adaboost.

In this work we describe a method in which the output of a face detector using brightness patterns is post-processed with the help of a skin detection method. We show that skin detection allows to remove many false positive detections, while maintaining the true positives. A major advantage of the approach presented here is that it employs only a small number of simple operations and thus can process images at a relatively high frame-rate. A further advantage is that - unlike many other methods - our algorithm returns the position of fiducial points, such as for example eye corners or mouth corners. This facilitates tasks such as face recognition or multimodal speech recognition using lip reading. Moreover, we show that the classification accuracy of our detector is competitive with the OpenCV implementation of the detector presented in (Viola and Jones, 2001).

The outline of the rest of this paper is as follows. In section 2 we describe face detection based on discrete Gabor jets (DGJ). Then, in section 3 we describe how color information can be used in a probabilistic model for face detection. In section 4 the method for combining results from the DGJ and color-based methods is explained. Finally, in section 5 results are presented and discussed.

## 2 FACE DETECTION USING DISCRETE GABOR JETS

The main idea underlying DGJ-based face detection is to first detect fiducial points such as eye corners and mouth corners and then to detect faces by verifying the relative positions of fiducial points with a reference graph. An overview of the different steps in the DGJ face detection process is given in Fig. 1.

First, edge detection is performed using a Canny edge detector with the Sobel operator. The goal of performing edge detection is to reduce the number of pixels that have to be analyzed and to focus on interesting non-uniform regions in the input image. After edge detection, features are extracted from the neighborhood of each edge pixel using rings of small rectangles as shown in Fig. 2. More precisely, a Fourier analysis is performed on single rings and on the contrast between adjacent rings. The feature vectors are then fed into a modified linear discriminant analysis (LDA) classifier, which allows to assign each edge pixel to one of the following seven classes: left or right eye corner, left or right nose corner, left or right mouth corner, and non-face fiducial point. The fiducial points are then combined to form face candidates using a reference graph. Finally, nearby face candidates are merged to avoid multiple detections of single faces.

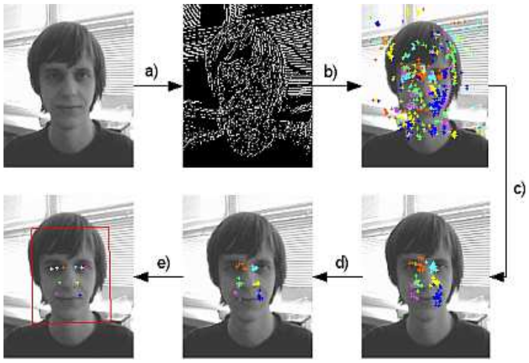


Figure 1: Face detection scheme: a) edge detection, b) LDA for discrete Gabor jets, c) facial features matching, d) merging facial features (separately, once for every scale), e) computing eye centers.

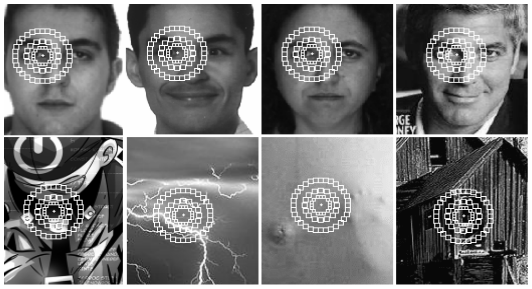


Figure 2: Rings of small squares as neighborhoods of analysis.

The methods for feature extraction, classification, and reference graph matching are described in more detail in the following.

### 2.1 Feature Extraction I: Discrete Gabor Jets

The Gabor filter (Gabor, 1946) in the spatial image domain is a Gaussian modulated 2D sine wave grating with parameters controlling wave front spatial orientation, wave frequency and rate of attenuation. While Gabor filters can be used to accurately represent local patterns with complex textures, the associated computational requirements exclude real time applications.

To allow for real time face detection, we use an efficient representation which describes changes of local image contrast around a given pixel in angular and radial directions. In particular, rings of small squares of pixels are used and the frequency of luminance changes on such rings is computed. Each single square is treated as one value by computing the sum of the luminance values of all the pixels that lie inside the square. The advantage of such an approach is that is computationally very efficient. In fact, the sum of the luminance values in a square can be easily computed (performing only two additions and two subtractions, no matter what is the size of the square) using integral images as proposed for the AdaBoost face detector (Viola and Jones, 2001).

We define two types of discrete Gabor jets (for details about the term “jet” see (Lades et al., 1993)). The first type detects angular frequencies on single rings, while the second type detects angular frequencies for the radial contrast between two rings with the same number of elements.

**Type 1 Jets.** Each jet of the first type is characterized by the radius $r$ of the ring, the number of squares $n = 2^k$ and the center (anchoring) point $(x,y)$. The sizes of all the squares on the ring are equal.

The sequence of the $n$ luminance values corresponding to the squares is normalized in order to be included in the unit interval $[0,1]$. This ensures robustness to illumination changes. Finally, the sequence of normalized luminance values is transformed with DFT. Only the first $n/2$ of the complex DFT coefficients are joined to the output feature vector. The mean value (DC) from the DFT is excluded from the feature vector.

**Type 2 Jets.** This type of jet consists of two rings with radii $r_1 < r_2$ with the same center $(x,y)$ and with the same number $n = 2^k$ of squares.

In contrast to the type I jets, now the mean value of each square is analyzed. This is done in order to compensate for differences in the size of squares in the inner and outer ring of the jet. Differences are taken between the mean values from the inner ring and the mean values from the corresponding outer ring. Next, the obtained differential signal is normalized to the unit interval and then transformed by DFT. Again, only the first $n/2$ of DFT complex coefficients are joined to the output feature vector. In contrast to the type I jets, the mean value (DC) is also included in the feature vector.

Table 1: Parameters of the discrete Gabor jets used in this work. Shown are the number of squares ($n$) used in the Gabor jets, and the radii ($r_1$, $r_2$) of the rings in pixels. The radii correspond to faces with a distance of 45 pixels between eye-centers and are scaled up or down to detect faces with bigger or smaller inter-eye distance.

| Type | 1 | 1 | 1 | 1 | 2 | 2 |
|---|---|---|---|---|---|---|
| $n$ | 16 | 16 | 32 | 32 | 16 | 32 |
| $r_1$ | 16 | 24 | 12 | 19 | 16 | 12 |
| $r_2$ | – | – | – | – | 24 | 19 |

To detect fiducial points at different scales, the radii of the rings are scaled up by steps of 1.15 until the rings become bigger than the input image. The size of the squares in the rings is adapted such that the squares are as big as possible but do not overlap. The exact parameters of the feature extractors used in this work can be found in Table 1.

## 2.2 Feature Extraction II: Modified Linear Discriminant Analysis

To compute a low-dimensional representation of the DGJ feature vectors, a modified version of LDA (Hotta et al., 1998) is used. The motivation for using a modified version of LDA is that with classical LDA it is difficult to achieve good separation between face fiducial points and non-face fiducial points. The reason for this difficulty seemingly is that the distributions of the face and non-face classes are very different, in particular one can expect the non-face class to have a much larger variance than the face class. Therefore, in this work we employ a version of LDA in which the concepts of within-class and between-class variance and related scatter matrices are modified. As a side effect, the modified version of LDA we are using allows to obtain vectorial discriminative features, even in the case of two-class problems.

Classical LDA maximizes the ratio of between-class variance $var_b$ to within-class variance $var_w$ (Fisher, 1936; Fukunaga, 1992):

$$f_X = \frac{\mathrm{var}_b(X)}{\mathrm{var}_w(X)}$$

$$\mathrm{var}_b(X) = \|\bar{x}^f - \bar{x}\|^2 + \|\bar{x}^{\bar{f}} - \bar{x}\|^2 \tag{1}$$

$$\mathrm{var}_w(X) = \frac{1}{|I_f|}\sum_{i\in I_f} \|x_i - \bar{x}^f\|^2 + \frac{1}{|I_{\bar{f}}|}\sum_{i\in I_{\bar{f}}} \|x_i - \bar{x}^{\bar{f}}\|^2,$$

where the training set $X$ of feature vectors is divided into the face class indexed by $I_f$ and the non-face class with remaining indices $I_{\bar{f}}$.

The non-face class is very differentiated, and, therefore, it is difficult to minimize its within-class variance. To solve this problem, we modify the expression for the within-class variance such that it only takes into account the face class. The expression for the between-class variance is modified such that the non-face examples are placed as far as possible from the center of the face-class. This leads to the following equations for the modified within-class and between-class variance:

$$m_X = \frac{\mathrm{mvar}_b(X)}{\mathrm{mvar}_w(X)}$$

$$\mathrm{mvar}_b(X) = \|\overline{x}^f - \overline{x}\|^2 + \frac{1}{|I_{\bar{f}}|}\sum_{i \in I_{\bar{f}}} \|x_i - \overline{x}^f\|^2 \qquad (2)$$

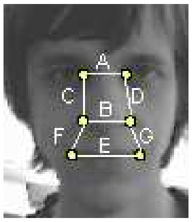
$$\mathrm{mvar}_w(X) = \frac{1}{|I_f|}\sum_{i \in I_f} \|x_i - \overline{x}^f\|^2.$$

The actual optimization procedure that allows to compute discriminant vectors from training data is very similar to the classical approach (for more details see (Hotta et al., 1998; Naruniec and Skarbek, 2007)).

## 2.3 Classification and Postprocessing

The feature extraction and dimension reduction steps described above result in $l$-dimensional feature vectors, where for each edge pixel we have one feature vector. To classify the edge pixels as eye corner, nose corner, mouth corner, or non-face, the euclidean distance $d$ of the feature vector $x$ to the centroid of each of the seven classes is computed. If the distance $d$ to the centroid $c$ of a given class is lower than a specified threshold, the edge pixel is classified as belonging to the corresponding class. The distance thresholds for separate facial features are tuned to the equal error rate (equal values of false rejection rate and false acceptance rate) during the training process.

After classifying all of the edge points as belonging to one of the seven classes, reference graph matching is performed. The reference graph consists of mean distances between the chosen fiducial points and has been computed using a set of 100 face images. All of the relations have been measured assuming, that the distance between the inner eye corners is always equal to 1 (see Fig. 3). To perform the graph matching, for each scale all possible combinations of the classified facial features are fitted to the reference distances. If the likelihood of the analyzed set of points, given the reference graph, is high, this set is marked as a face. To avoid checking all of the combinations, some preliminary checks may be performed. For example one can check if the left eye is on the left side of the right eye, or if the nose is lower than the eyes.



| edge | length |
|---|---|
| A | 1.00 |
| B | 1.05 |
| C | 1.02 |
| D | 1.02 |
| E | 1.5 |
| F | 0.85 |
| G | 0.85 |

Figure 3: Fiducial points and their distance relations used in distance matching algorithm.

In the next step, the closest results are merged to avoid multiple detections. Finally, outer eye corners of the found faces are searched within the closest neighborhood of the inner eye corners. The eye center position is placed in the middle of the inner and outer eye corners.

For more details about the classification and postprocessing steps, see (Naruniec and Skarbek, 2007; Naruniec et al., 2007).

# 3 FACE DETECTION USING COLOR INFORMATION

As was shown in previous publications (Naruniec and Skarbek, 2007; Naruniec et al., 2007), face detection using DGJs allows to achieve very good results, i.e. a large number of faces present in the test images is detected, while non-faces are only rarely accepted as faces. However, DGJ-based face detection ignores a large part of the information contained in digital images, namely color information. Hence, by making use of color information it might be possible to further improve the results of DGJ-based face detection. To make use of color information, the eye coordinates returned by the DGJ method are used to extract rectangular image patches containing face candidates. The face candidates are then verified using color information and probabilistic models of image patches.

The details of the color-based face detection method are described in the following sections, starting with a description of probabilistic models for skin color and other colors. The steps that are necessary to combine DGJs and the color-based method are described in section 4.

### 3.1 Modeling Skin Color and Other Colors

The main idea underlying our probabilistic color-based face detection method is to describe images of frontal upright faces as a mixture of pixels that have skin-like color and of pixels that have other colors. To model the distributions of skin color and other (non-skin) colors, the approach described in (Jones and Rehg, 2002) is used. More specifically, the distributions of skin and non-skin color are learned from the database described in (Jones and Rehg, 2002). The database contains nearly 1 billion pixels labeled as skin or non-skin and thus allows to build a skin detector which is relatively robust to skin color variations caused by variations in ethnicity, scene illumination, or camera characteristics.

To model the distributions of skin and non-skin color, three-dimensional histograms of size 16×16×16 are used, i.e. the 256 possible values of the R, G, and B channels are quantized into 16 equally spaced bins. Learning the distribution of skin (non-skin) color then corresponds to counting the number of pixels labeled as skin (non-skin) for every bin of the histogram and dividing by the total number of pixels labeled as skin (non-skin). The result of training are a vector of probabilities $\theta^s$ for the skin color histogram and a vector of probabilities $\theta^o$ for the non-skin color histogram. For each bin in the skin and non-skin histograms, the probability vectors describe how probable it is that a pixel has a combination of R, G, and B values drawn from that bin. Now, the color $\mathbf{c} = \{R, G, B\}$ of any pixel can be modeled as a mixture of the distributions of skin color and non-skin color:

$$p(\mathbf{c}|\theta^s, \theta^o, \pi) = p(\mathbf{c}|\theta^s)\pi + p(\mathbf{c}|\theta^o)(1-\pi). \quad (3)$$

Here, the probability $\pi$ is used to describe how probable it is a priori that a pixel has skin color. Using this model, the posterior probability for skin can be computed using Bayes rule.

$$p(\text{skin}|\mathbf{c}, \theta^s, \theta^o, \pi) = \frac{p(\mathbf{c}|\theta^s)\pi}{p(\mathbf{c}|\theta^s)\pi + p(\mathbf{c}|\theta^o)(1-\pi)} \quad (4)$$

Typical results of computing skin probability for some color images are shown in Fig. 4.

Note that computing skin probability maps can be done relatively fast. To compute an index into the skin and non-skin histograms, two integer additions and two integer multiplications are necessary for each pixel. After looking up the values corresponding to the index in the histograms, two floating point multiplications, one floating point addition, and one floating point division are necessary. In summary each pixel thus requires only four integer operations, four floating point operations, and two memory accesses, making the computation of skin probability maps at high framerates feasible.

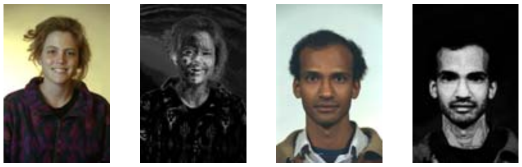


Figure 4: Examples for skin detection. Left: original image, right: results of skin detection. Bright pixels represent high skin probability, dark pixels represent low skin probability. As can be seen skin detection with color histograms yields good results independently of ethnicity.

While it would be possible to directly use skin probabilities, for example to filter out regions not containing faces, we have developed a more powerful approach, which allows to model the shape of regions containing skin color. This approach is described in the following.

### 3.2 Modeling Colored Image Patches

Assuming that the color of pixels at different positions is independent, the probability for observing colors $\mathbf{c}_i$ and $\mathbf{c}_j$ at positions $i$ and $j$ is

$$p(\mathbf{c}_i, \mathbf{c}_j|\theta^s, \theta^o, \pi) = p(\mathbf{c}_i|\theta^s, \theta^o, \pi)p(\mathbf{c}_j|\theta^s, \theta^o, \pi). \quad (5)$$

Now the probability for observing an image patch with $N$ pixels can be expressed as follows:

$$p(\mathbf{c}_1, \ldots, \mathbf{c}_N|\theta^s, \theta^o, \pi) = \prod_{i=1}^{N} p(\mathbf{c}_i|\theta^s, \theta^o, \pi_i). \quad (6)$$

Here we have slightly changed the notation to express the fact that the mixture coefficients depend on the position of pixels within an image patch. The model for image patches with $N$ pixels is thus fully specified by a vector of $N$ mixture coefficients $\pi = \{\pi_1, \ldots, \pi_N\}$ and by the histogram parameters $\theta^s$ and $\theta^o$.

### 3.3 Learning Parameters of Image Patch Model

We use the skin and non-skin color histograms $\theta^s$ and $\theta^o$ learned from the dataset described in (Jones and Rehg, 2002) (see previous section). Learning parameters of an image patch model then just corresponds to learning the mixture coefficients $\pi_i$ for every pixel in the patch.

Given a training set of image patches, the mixture coefficients are computed with a simple maximum-

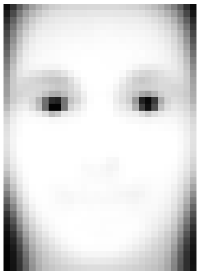


Figure 5: Mixture coefficients of the model for face image patches. Bright pixels indicate a high probability for skin, dark pixels indicate a low probability for skin.

likelihood method. Denoting the color values of training image $j$ at position $i$ by $\mathbf{c}_{ij}$, the number of training patches by $M$, and the collection of all training patches by $\mathbf{D}$, the likelihood for the mixture coefficients $\pi = \{\pi_1, \ldots, \pi_N\}$ is

$$p(\mathbf{D}|\theta^s, \theta^o, \pi) = \prod_{j=1}^{M}\prod_{i=1}^{N} p(\mathbf{c}_{ij}|\theta^s, \theta^o, \pi_i). \qquad (7)$$

The log-likelihood then is

$$\log p(\mathbf{D}|\theta^s, \theta^o, \pi) = \sum_{j=1}^{M}\sum_{i=1}^{N} \log p(\mathbf{c}_{ij}|\theta^s, \theta^o, \pi_i). \quad (8)$$

Taking the partial derivative of the log-likelihood with respect to $\pi_i$ we obtain

$$\frac{\partial \log p(\mathbf{D}|\theta^s, \theta^o, \pi)}{\partial \pi_i} = \sum_{j=1}^{M}\sum_{i=1}^{N} \frac{p(\mathbf{c}_{ij}|\theta^s) - p(\mathbf{c}_{ij}|\theta^o)}{p(\mathbf{c}_{ij}|\theta^s, \theta^o, \pi_i)}. \qquad (9)$$

Finally, to maximize the log-likelihood, the partial derivatives are used to perform gradient ascent until convergence. In our experience this method for maximizing the log-likelihood converges fast and reliably. The result of computing the mixture coefficients for the face class is shown in Fig. 5.

### 3.4 Face Detection using the Image Patch Model

To perform face detection with the image patch model, two sets of mixture coefficients are used. One set, denoted by $\pi^f$, is learned from a large number of image patches containing faces. The other set, denoted by $\pi^o$, is learned from a large number of image patches not containing faces. Now, given an image patch $\mathbf{P}$ and a prior probability $p(f)$ for faces, Bayes rule can be used to compute the probability that the patch contains a face:

$$p(f|\mathbf{P}, \pi^f, \pi^o, \theta^s, \theta^o) \qquad (10)$$
$$= \frac{p(\mathbf{P}|\pi^f, \theta^s, \theta^o)p(f)}{p(\mathbf{P}|\pi^f, \theta^s, \theta^o)p(f) + p(\mathbf{P}|\pi^o, \theta^s, \theta^o)(1-p(f))}$$

We decide that the image patch contains a face if the posterior probability is bigger than a threshold $\tau$.

## 4 COMBINING DISCRETE GABOR JETS AND COLOR INFORMATION

The two methods are combined by first running face detection using discrete Gabor jets (DGJ) on a grayscale version of the input image. Then, the face candidates found by the DGJ method are verified with the help of the color based method described in section 3.

More specifically, a bounding box is computed from the coordinates of left eye and right eye as returned by DGJ. To compute the width $w$, height $h$, and coordinates $x, y$ of the top left corner of the bounding box, the following equations are used:

$$w = (x_r - x_l) f_w \qquad (11)$$

$$h = \frac{w h_s}{w_s} \qquad (12)$$

$$x = \frac{x_l + x_r}{2} - \frac{w}{2} \qquad (13)$$

$$y = \frac{y_l + y_r}{2} - h f_y \qquad (14)$$

Here $x_l, y_l$ and $x_r, y_r$ denote the coordinates of the pupils of the left and right eye returned by the DGJ method. The anthropometric constants $f_w$ and $f_h$ are set to 2.1 and 0.37. The width $w_s$ and height $h_s$ of the standard size face patch are set to 30 pixels and 40 pixels.

After computation of the bounding box, the corresponding rectangular region is extracted from the input image and resized to standard size. To resize patches bigger than the standard size, lowpass filtering followed by bilinear interpolation is used. To resize patches smaller than the standard size, only bilinear interpolation is used. Face candidates found by DGJ are accepted as true faces if the posterior probability for face (see equation 10) is bigger than a threshold $\tau$. The threshold can be used to vary the characteristics of the combined detector and to create ROC curves.

## 5 RESULTS

The combined face detector was tested on images from the BANCA and VALID databases (Fox et al., 2005; Bailly-Bailliére et al., 2003). For testing, first, the DGJ method was applied to compute a set of face

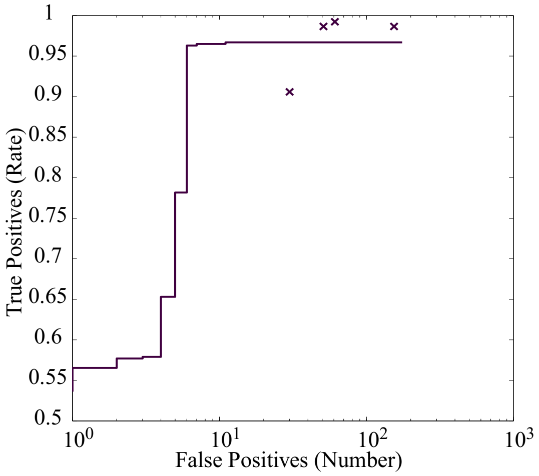


Figure 6: ROC curve for face detection on the BANCA database with a combination of DGJ and color based face detection. The rightmost point on the ROC curve corresponds to the result achieved with DGJ alone, while the other points on the ROC curve where computed by combining DGJ with the color based method. The crosses represent the results obtained by OpenCV detector.

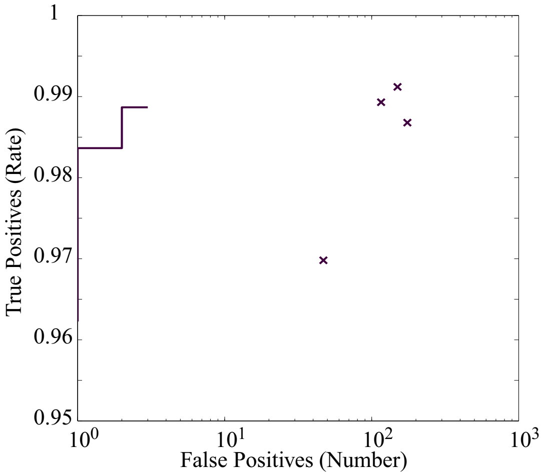


Figure 7: ROC curve for face detection on the VALID database with a combination of DGJ and color based face detection. The rightmost point on the ROC curve corresponds to the result achieved with DGJ alone, while the other points on the ROC curve where computed by combining DGJ with the color based method. The crosses represent the results obtained by OpenCV detector.

candidates for each input image. Each face candidate was then verified with the help of the image patch model from section 3. By varying the threshold for acceptance, ROC curves were created.

In order to compare the performance of our combined face detector with a standard face detector, the OpenCV implementation of the detector presented in (Viola and Jones, 2001) was employed. In the OpenCV implementation there are four classifiers, namely, default, alt, alt2 and alttree. Table 2 shows the performance of the OpenCV detectors for the BANCA and VALID datasets.

The ROC curve for the BANCA database is shown in Fig. 6 and the ROC curve for the VALID database is shown in Fig. 7. The crosses in these figures represent the true positive rate and number of false positives achieved with the OpenCV detector.

In both figures the rightmost point on the ROC curves corresponds to using a threshold of 0, i.e. all face candidates computed with the DGJ method are accepted. As can be seen, higher thresholds allow to reject many false positives detected by DGJ, while keeping almost all true positives. The results are especially striking for the BANCA database, where the color information allowed to reduce the number of false positives from about 180 to less than 20, without loosing any true positives. For the VALID database the DGJ detector returned only three false positives but nevertheless the color information allows to remove these while keeping almost all true positives.

Comparing the ROC curves with the crosses in Fig. 6, it can be perceived that although three of the OpenCV detectors have a slightly better true positive rate than our detector, they bring about a high number of false positives. It can be also seen that the true positive rate deteriorates when an OpenCV detector with a smaller number of false positives is employed. On the contrary, combining DGJ-based and color based face detection, leads to a considerable decrease in the number of false positives while maintaining the true positive rate.

For the VALID database, as it can be observed in Fig. 7, the DGJ detector achieves a considerably small number of false positives and nearly the same true positive rate as the OpenCV detector. Interestingly, it can be observed that even the small number of false positives resulting from using DGJ can be reduced to zero, when color information is used for classification.

## 6 CONCLUSIONS

In this work, an efficient face detection system was presented. Firstly, the DGJ detector was employed for face detection based on the brightness information of the pixels in images. In the next step, color information of the pixels was used to post-process the results

Table 2: Number of false positives (FP) and false negatives (FN) for the BANCA dataset (520 images) and for the VALID dataset (1590 images).

| | BANCA | | VALID | |
|---|---|---|---|---|
| | FP | FN | FP | FN |
| default | 154 | 7 | 175 | 21 |
| alt2 | 61 | 4 | 150 | 14 |
| alt | 51 | 7 | 116 | 17 |
| alttree | 30 | 49 | 47 | 48 |

obtained by DGJ detector. To this end, colored image patches were modeled and this model was employed in the final decision-making about whether the face detected by DGJ is true or false positive. The results have shown that employing features related to color information and combining them with brightness information will lead to a considerable decrease in the number of false positives, while maintaining the true positive rate. Consequently, using the system introduced in this paper, will lead to a higher correct classification rate and face detection accuracy.

## ACKNOWLEDGEMENTS

The work presented was developed within VISNET II, a European Network of Excellence (http://www.visnet-noe.org), funded under the European Commission IST FP6 Programme. The authors wish to express their thanks to this network of excellence.

## REFERENCES

Bailly-Bailliére, E., Bengio, S., Bimbot, F., Hamouz, M., Kittler, J., Mariéthoz, J., Matas, J., Messer, K., Popovici, V., Porée, F., Ruiz, B., and Thiran, J.-P. (2003). The BANCA database and evaluation protocol. In *Audio- and Video-Based Biometric Person Authentication*, volume 2688 of *Lecture Notes in Computer Science*.

Feraud, R., Bernier, O., Viallet, J.-E., and Collobert, M. (2001). A fast and accurate face detector based on neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):42–53.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188.

Fox, N. A., O'Mullane, B. A., and Reilly, R. B. (2005). VALID: A new practical audio-visual database, and comparative results. In *Audio- and Video-Based Biometric Person Authentication*, volume 3546 of *Lecture Notes in Computer Science*.

Fukunaga, K. (1992). *Introduction to Statistical Pattern Recognition.* Academic Press.

Gabor, D. (1946). Theory of communication. *Journal of the Institute of Electrical Engineers*, 93(3):429–457.

Hotta, K., Kurita, T., and Mishima, T. (1998). Scale invariant face detection method using higher-order local autocorrelation features extracted from log-polar image. In *International Conference on Face & Gesture Recognition*, page 70.

Hsu, R.-L., Abdel-Mottaleb, M., and Jain, A. (2002). Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):696–706.

Huang, S.-H., Huang, S.-H., and Lai, S.-H. (2004). Detecting faces from color video by using paired wavelet features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 64–64.

Jones, M. J. and Rehg, J. M. (2002). Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96.

Lades, M., Vorbrüggen, J., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R., and Konen, W. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, 42:300–311.

Naruniec, J. and Skarbek, W. (2007). Face detection by discrete gabor jets and reference graph of fiducial points. In *Rough Sets and Knowledge Technology*, volume 4481 of *Lecture Notes in Computer Science*.

Naruniec, J., Skarbek, W., and Rama, A. (2007). Face detection and tracking in dynamic background of street. In *International Conference on Signal Processing and Multimedia Applications (SIGMAP)*.

Rowley, H., Rowley, H., Baluja, S., and Kanade, T. (1998). Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38.

Sung, K.-K. and Poggio, T. (1998). Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 511–518.

Yang, M.-H. and Ahuja, N. (1998). Detecting human faces in color images. In *International Conference on Image Processing (ICIP)*, pages 127–130.

Yang, M.-H., Kriegman, D. J., and Ahuja, N. (2001). Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:34–58.