A *Pattern Matching* Method and Algorithm for *Face Detection*

Mihnea Horia VREJOIU

National Institute for R&D in Informatics – ICI Bucharest
8-10 Averescu Blvd., Bucharest, 011455, Romania.
mihnea@dossv1.ici.ro

Abstract: The paper presents a simple, original method, using supervised learning and pattern matching, for frontal-view face detection. The raw method and algorithms, their refinements and optimizations, the experimental system, and the obtained results are described incrementally. A pyramid of several simplified representations for faces, with gradual complexity and dimensionality from coarser to more detailed ones, has been defined. These representations were used as a basis for structuring and organizing the knowledge base as a kind of hash tree, as well as to optimize the developed algorithms and their processing time, by applying a sequence of filters with gradual computational complexity in cascade at detection. Appropriate metrics have been defined and used to evaluate the similarity between two finite sets of binary values, of the same dimension, with a minimum of computational effort in these filters. Various thresholds of passage were empirically chosen and adjusted. An experimental system was developed and used for learning and detection tests. Public image databases and private images have been employed, and quite promising results have been obtained. Finally, comparative parallels with some reference methods are discussed.

Keywords: face detection, supervised learning, positive / negative examples, hash code, clusterization, pattern matching, similarity score, cascade of filters.

1. Introduction

The ways human faces appear in digital images – still photos, video frames – cover a very wide variety. And this, not necessarily due to individual peculiarities, but mainly due to the way light and shadows are playing on the relief (3-D shape) of the respective faces, depending on the illumination conditions. On the other hand, the dimensions and resolutions of the images might be different, and the positions in which human faces may appear within them as location, framing, relative and absolute size, slant and/or rotation angles, are very numerous. Due to all the above-mentioned factors, the problem of human faces detection in images appears as a difficult one. Innovative researches and their outcomes in the last decades [22][25], mostly based on various representations and learning from examples techniques, brought remarkable contributions and results in the field worldwide, both as the rate of success for the detection methods and algorithms, and as processing speed (see also section 5). Today, effective face detection is hard-coded in almost all digital cameras, smartphones and tablets.

This paper presents an attempt of an original, simple approach for face detection, in frontal view as a first phase. The starting idea was to firstly bring the human faces at certain representation forms as much as possible less dependent on peculiarities as mentioned above. Also, the proposed method was envisaged to be as general as possible, without constraints, tolerant at scaling, translation, slight skews or rotations, independent on the image type (color or grayscale), and as efficient as possible as volume and as duration of the computations involved. Next sections present the raw method and algorithm (section 2), and their refinement and optimization (section 3). Section 4 describes the experimental system developed, and some of the results obtained. In section 5, comparative parallels with some reference methods and algorithms are made, while section 6 presents some conclusions and possible future works.

2. Raw method and algorithm

The considered starting input is a digital image with 256 gray levels, pixel per byte. In the case of color images, appropriate conversion to grayscale [4] is performed. For example, the transformation from the RGB model, with each color coded on one byte, is based on the NTSC luminance (panchromatic brilliance) relation, which uses coefficients based on the human eye sensitivity for each of the three components:

$$GL = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B, \qquad (1)$$

where *GL* is the value of the correspondent gray level pixel. In the case of the YUV/Y'UV or YCbCr/Y'CbCr representations, with one brilliance component (*Luma / Luminance*) and two color components (*Chroma / Chrominance*), or HSV/HSI/HSL, with *Hue*, *Saturation* and *Value / Intensity / Luminance* components, the

values from the last component, Y/Y' or V/I/L respectively, could be directly considered for the grayscale image.

In the input digital images with 256 gray levels, human faces are represented by the spatial configuration of the pixels with various gray level values, compounding the respective faces. Depending to the number of those pixels and their values, a huge variety of configurations may appear, and they must be automatically identified as faces. It appears almost natural to try to use essentialized representations of these faces, with reduced dimensionality but intrinsic power of generalization, and to use an approach based on learning first such representations from several examples of faces, and then detect similar representations in (other) images, corresponding to (other) human faces.

Based on the observation that the relevant information for a human face are contained within an approximately square region bounded by the eyebrows and mouth, and for ensuring invariance at scaling, it has been chosen to use square representations with unique dimensions to which any human face would be normalized. A face area is framed with a flexible square frame placed and adjusted to properly cover the entire region from eyebrows till mouth, inclusively. This framed area is split in N×N square *macro-pixels*, each being computed as the average of the gray levels of the pixels from the original image covered by the respective *macro-pixel*. In order to ensure both an enough resolution in the case of the smallest detectable faces and computational efficiency at same time, a value of N = 21, divisible by 3 (see RBM in Section 3) has been chosen as tradeoff. This normalized 21×21 square representation could be referred as *matrix of gray levels*, MG.

Then, for obtaining a certain generalization and independence on local details and noises, a smoothing operation is applied on MG using a *mean filter* [4], by which the value of each *macro-pixel* is replaced with the average value of its 8-neighbors. Actually, two convolutions with a 3×3 matrix having all its components 1/8, excepting the central one, which is 0 (zero), are performed consecutively on MG. An effect of blurring, or viewing through a matte glass, is obtained, details and contrast being thus reduced, while the characteristic, essential big elements, which are defining the respective face image, remain (middle row in Image 1). The new 21×21 matrix, resulted from the two convolutions will be referred as *medium gray levels matrix*, MGM.

We mention here the fact that, based on this representation, a local gradient, due to the effect of light and shadows on the relief (3-D shape) of the face, was also evaluated and used, as detailed in section 2.

For avoiding possible influences of hair (in the region of the forehead, eyebrows and eyes) or of background (in the region of the jaws and neck) on the faces, we shall further mask the corners of the 21×21 square matrix MG, as isosceles triangles with sides length of 7 in the lower half and of 3 in the upper one, as well as its first and last 2 lines and columns.

The MGM representation is then binarized, by using an adaptive threshold value computed based on the histogram of the gray levels [4][5] of its *macro-pixels* that are not masked. A new 21×21 matrix is obtained, with values of 0 and 1 (black and white). Some black "stains" – *patterns* – on a white background appear in these binarized matrices, having various shapes and dimensions, and corresponding to the darkest areas of the respective face (most of them due to the shadowed regions, eyes, eyebrows, nostrils, mouth, beard and/or moustache). For an even greater generalization, and for more invariance at slight scaling, translation and rotation, a simplified form of morphological *dilation* is also performed on these black patterns, finally resulting a new representation, referred as *binary matrix*, BM.

Image 1. MGM (middle row) and BM (lower row) representations, for slightly different framing and positions of a face from an image with 256 gray levels (upper row)

By simplifying things for the moment, we may consider that, through a *supervised learning* from examples process, such 21×21 binary matrices

(BM) representations of the faces are collected in a *knowledge base* (KB). This KB may be further used with a *pattern matching* algorithm for identifying in (other) images regions which lead to similar representations, and thus to detect (other) faces in these ones. In the following we shall briefly describe this algorithm. The 256 gray levels input image, of W×H pixels, is systematically scanned with a flexible, square window, of 21×21 pixels initially, starting from the upper-left corner and sliding over it with a certain step *p = 1, 2, or 3* pixels, horizontally and vertically, "row after row", in consecutive positions until the entire image is thus covered. Then, the sides of the square window are scaled either by a factor (of *1.10:1 – 1.25:1*) or by adding a constant number of pixels (e.g. *3, 5, 7*) at each new scale, and the scanning of the target image is started again with this new window. This process continues until the scanning window can't be scaled anymore because its sides would exceed the target image borders (W and/or H). At each scale and for each position of the scanning window, a normalized representation MG_t of 21×21 *macro-pixels* is first generated from the area of the input image covered by the target window at the respective step. The smoothing filter is then applied on this MG_t, resulting a *medium gray levels matrix*, MGM_t. By applying binarization and dilation on this one, a *binary matrix* BM_t is finally obtained. This latter one is then compared successively with all the examples BM_i stored in the *knowledge base*, with which the system was previously trained. For each comparison, a similarity score, $0 \le s_i \le 100$, is computed, proportional with the resemblance degree between the target representation and the model representation "i". If the maximum score $S_i = max(s_i)$, obtained after comparing BM_t with all the BM_i representations from the *knowledge base*, is greater than a threshold *b_thr*, empirically chosen through experiments, the respective target window is collected as possible candidate to contain a face, together with the respective maximum score, S_i.

For evaluating the similarity score s_i between two 21×21 representations corresponding to the target binary matrix, BM_t, and respectively to the model binary matrix "i", BM_i, previously learned as a positive or negative example, there have been defined and are computed, using the relations (2) and (3) given below, two values that quantify, each, the similarity degree between the two matrices. Be n_t and n_i the total number of black pixels (with value 0) in BM_t and in BM_i respectively, $n_{ti} = n_t + n_i$, the total number of black pixels in both these matrices, and nd_{ti} the total number of black pixels in both these matrices that don't have a correspondent in the other one (XOR). A similarity degree based on the "*fullness*", $0 \le Sf_i \le 1$, and a similarity degree based on the "quantity of differences", $0 \le Sd_i \le 1$, are given by:

$$Sf_i = 1 - min\left(1, \frac{2 \cdot | n_t - n_i |}{n_{ti}}\right), \quad (2)$$

$$Sd_i = 1 - min\left(1, \frac{2 \cdot nd_{ti}}{n_{ti}}\right). \quad (3)$$

It could be observed that, if n_t is close to n_i, namely in the situation that the target and model have approximately the same number of black pixels, regardless the way these ones are spatially distributed in the two binary matrices, then Sf_i is approximately 1 (even 1 in case of equality). In the cases with big differences between the numbers of black pixels that are giving the "*fullness*" of the two binary matrices, Sf_i gets smaller values (decreasing to 0 when either the target or the model contains less than 1/3 black pixels than the other one). Similarly, Sd_i tends to a value of 1 if there are only very few black pixels in either one of the two matrices which don't have a correspondent in the other one, which is the case of almost identical matrices, while it's decreasing to a value of 0 for a number of differences greater than the average number of black pixels in each of the two matrices. Sd_i gives the measure in which the distributions of black pixels in the two matrices are similar as spatial localizations.

Finally, the *similarity score* is defined and computed as the product of the above two degrees of similarity, as percent:

$$s_i = 100 \cdot Sf_i \cdot Sd_i \, \%. \quad (4)$$

One may remark that relation (3) defining the similarity degree based on the quantity of differences between the two binary matrices that are compared, may be seen as an adapted complementary form of *Tanimoto's similarity degree* [15], as *Jaccard index / coefficient* [9], respectively *Sørensen-Dice coefficient / index* [1][18], for the case of two binary sets of the same dimension. Experimentally, it has been found that, instead of computing the *similarity score* through the sequence of relations (2)÷(4), this one might be directly computed –

with more or less comparable results if appropriately adjusting threshold parameters – as a particular variant of *Jaccard coefficient* (defined as ratio between the dimensions of intersection and reunion of two finite sets), which evaluates the distributions of the black pixels in the two binary matrices. Be nc_{ti} the number of the black pixels that are located at the same coordinates in the two compared binary matrices, nd_{ti} the total number of the black pixels in these matrices that don't have a correspondent in the other one (XOR), through which they are differing, and $n_{ti} = nc_{ti} + nd_{ti}$, the total number of the black pixels in both matrices. The similarity score, as percent, is given by:

$$s_i = 100 \cdot \frac{nc_{ti}}{n_{ti}} \%. \qquad (4')$$

We mention that, after computing each of the similarity degrees with the relations (2) and (3), if the respective one is lower than the threshold *b_thr*, further comparison with current model "i" is skipped. If all the models in the *knowledge base* are thus skipped, the current target window is rejected, and the algorithm goes to the next position or scale. One may consider these as a *cascade of filters* applied during the analysis of the current window.

Due to the intrinsic potential for generalization of the binary representation that we used, often, mostly for big faces, for a same face not one, but several overlapped candidate windows with dimensions and/or positions slightly different between them are obtained. Also, depending on the value chosen for the *b_thr* threshold, it is expectable to get several or fewer false detections, too. A smaller *b_thr* threshold allows a more optimistic generalization, but with the drawback that it will lead also to more *false-positives*, while a higher *b_thr* threshold will restrain the generalization, but will also diminish the number of false detections. The key here consists in finding the optimum tradeoff between the two alternatives, the appropriate adjustment of the threshold being realized through experiments, trials and testing.

During the training process, both positive and negative examples may be learned from normalized 21×21 framing windows effectively containing faces, or respectively, that don't contain faces, each being appropriately marked with a tag (T) in the *knowledge base*. We are talking about an interactive, user guided, *supervised learning* mechanism.

The framing of each human face that should be learned as a positive example is performed manually, from the faces undetected yet after initially trying an automatic detection on the respective image using the existing experience (*knowledge base*), available at that moment. In the case that after the detection on the current image some *false-positives* appear too, these ones may also be manually selected one by one, and learned as negative examples.

3. Refinements and optimizations

Once the *knowledge base* starts growing, the comparison of all the representations obtained from each target window, scaled and glided step by step over the image, with all previously learned examples for detecting possible faces, becomes excessively expensive as duration. Also several *false-positives* appear. Therefore, the detection algorithm described in previous section needs refinements and optimizations.

Firstly, for improving detection performance as well as processing speed, some more coarse representations have been also generated. Based on them, at detection, corresponding filters, with gradual computing complexity, were added to the cascade. They allow early rejection, with minimum computational effort, of those target windows that are definitely not susceptible to be detected as containing a face (relying on the content of the *knowledge base* at the respective moment). Also, even the structure of the *knowledge base* has been refined and optimized based on these coarse representations, for minimizing the computing effort and duration for each position and scale of the scanning window.

On one hand, a *hash code* vector, HC, has been defined, whose 9 components conventionally quantify a coarse description of the *binary matrix* BM, as described in the following.

Be $n_1 \div n_4$ the number of black pixels (values of 0) in each quarter of the square 21×21 matrix BM, indexed starting from the upper half, left to right, $n = \sum n_i,\ i = 1 \div 4$, the total number of the black pixels in the whole matrix, and $n_{IJ} = n_I + n_J$, the total number of black pixels in the halves formed by the pairs of horizontally and respectively vertically adjacent quarters {I,J}, where $\{I,J\} = \{1,2\}, \{3,4\}, \{1,3\}, \{2,4\}$.

The first component, HC(1), quantifies in 6 conventional domains the "*fullness*" of matrix BM, based on the number of black pixels (values of 0) contained by it, as below:

If $n < N^2 / 7$, *then* $HC(1) = 0$;
else, if $n < N^2 / 3$, *then* $HC(1) = 1$;
else, if $n < N^2 / 2$, *then* $HC(1) = 2$;
else, if $n < 2 \cdot N^2 / 3$, *then* $HC(1) = 3$;
else, if $n < 6 \cdot N^2 / 7$, *then* $HC(1) = 4$;
else, $HC(1) = 5$; $(N^2 = 21^2 = 441)$. (5)

The second and third components, HC(2) and HC(3), quantify in 3 conventional categories the distribution of the black pixels among the two halves, on the vertical and respectively horizontal, in the binary matrix BM, coarsely reflecting the way they are balanced in these halves. Be $k = 0,6 \cdot n$, an empirically chosen value for ensuring a certain tolerance to possible noises, which represents 60% from the total number of the black pixels in the whole matrix. These components are thus defined:

If $n_{13} > k$, *then* $HC(2) = 1$;
else, if $n_{24} > k$, *then* $HC(2) = 2$;
else, $HC(2) = 0$, (6)

If $n_{12} > k$, *then* $HC(3) = 1$;
else, if $n_{34} > k$, *then* $HC(3) = 2$;
else, $HC(3) = 0$. (7)

The next six components, HC(4) ÷ HC(9), quantify, also in 3 conventional categories, the distribution of the black pixels among the pairs of quarters, on vertical, horizontal and, respectively, diagonal in the binary matrix BM, coarsely reflecting the way they are balanced in these quarters. Be $k_{IJ} = 0,6 \cdot n_{IJ}$, with $n_{IJ} = n_I + n_J$, an empirically chosen value for ensuring a certain tolerance to possible noises, which represents 60% from the total number of the black pixels in each pair {I,J} of quarters, where *{I,J} = {1,3}, {2,4}, {1,2}, {3,4}, {1,4}, {2,3}*. These HC(m) components, where *m = 4, 5, 6, 7, 8, 9*, each corresponding to one such pair of quarters {I,J} are defined as:

If $n_I > k_{IJ}$, *then* $HC(m) = 1$;
else, if $n_J > k_{IJ}$, *then* $HC(m) = 2$;
else, $HC(m) = 0$. (8)

This *hash code* vector, HC, is also stored in the *knowledge base* while learning, together with the binary matrix BM, and the tag T specifying the type of the example, positive or negative. Moreover, the *knowledge base* is structured as a kind of *hash tree*. On each branch all the representations with identical *hash code* vectors HC_i – as nodes of the respective tree – are gathered, positive and negative samples on separate branches. This allows the comparison of the target window only with those models from the branch with same HC_i as the current HC_t at detection, which leads to an obvious optimization in terms of computing duration and efficiency. Whether HC_t is different of all positive HC_i from the *knowledge base*, the current target window is directly rejected. We may consider this as another filter, HC, in cascade, preceding the BM one described in the previous section, which will be applied only to those windows that passed the HC filter.

Another coarse representation is obtained from the square binary matrix BM by reducing its dimensions to 1/3 for each of its sides, resulting a 7×7 *reduced binary matrix* (RBM). Each new *macro-pixel* in this one is obtained by replacing each non-overlapping block of 3×3 pixels from BM with a value of 0 if there exist at most 4 white pixels (values of 1) in the respective block, or with a value of 1 otherwise (Image 2).

Image 2. RBM representation of a face

At detection, while scanning the image, for each target window, exclusively on the branch of the *knowledge base* tree corresponding to the current HC_t, firstly the RBM_t will be compared with each RBM_i on the respective branch, using the same similarity score formulae as described for the BM in previous section. This may be considered also as another filter inserted in the cascade, between the HC and the BM ones, more efficient computationally than the latter one due to the smaller dimensions of RBM. Thus, only for those windows that passed the HC and RBM test, the BM comparison will be also performed, while the others are rejected.

It should be mentioned that a different threshold, *3_thr*, is used in the RBM case instead of *b_thr*, its value being also experimentally adjusted.

A remark should be made in the case of the negative examples learned in the *knowledge base*. When comparing a target window with this ones as described above, another specific threshold *n_thr* is used, whose value was also experimentally adjusted, any similarity score above this value leading to the rejection of that window in the early stages during detection.

Finally, for (better) filtering the *false-positives* resulted at detection, a local gradient related representation, based on the MGM one as mentioned in the previous section, was also introduced. It is a 21×21 *relief (3-D shape) matrix* (RM), whose components may get one of 3 conventional values a_1, a_2, a_3, (arbitrarily chosen *50, 200 and 125*), representing one of the categories: "valley", "peak" and "plateau" respectively, as below:

RM(x,y) = a_1, if over 4 of the 8-neighbors of pixel MGM(x,y) have greater values than it;
RM(x,y) = a_2, if over 4 of the 8-neighbors of pixel MGM(x,y) have smaller values than it;
RM(x,y) = a_3, in other cases. (9)

Obviously, the term "relief" and the values comparisons are referring to the gray levels, but these are however intimately related to the 3-D shape from the image window as it appears in the respective illumination conditions.

These RM representations (Image 3) quantify information about the local gradient around each *macro-pixel* in MGM, i.e. whether this one is lower, higher, or at the same level with the majority of its neighbors.

Image 3. RM representations for the face with slightly different framing and positions from Image 1

During the training process, such RM_i matrix is also stored in the *knowledge base* together with the other representations mentioned until now, for each learned example. At detection, for each target window that passed all the previous filters in cascade, a comparison of the RM_t and RM_i representations will also be performed. Finally, there will be collected only those candidates for which also the similarity score sr_i obtained in this case is greater than a threshold *r_thr*, also experimentally adjusted.

The *similarity score* sr_i is simply computed in this case as a *Tanimoto similarity degree* (*Jaccard coefficient*), respectively as the percent ratio between the number of components in RM_t which have equal values with their correspondent component in RM_i, and the total number of the compared pairs of components, avoiding the masked corners as shown in the previous section. It should be mentioned here that the comparisons are made in fact on 3×3 neighborhoods, for covering also slightly shifted but similar representations.

Concluding, complete pyramidal data structure of the representations for the examples of human faces learned in the *knowledge base*, containing the components described above, sorted based on their growing dimensions and the order of their correspondent similarity filter in the cascade at detection, is shown in Table 1.

Table 1. Pyramid of representation forms

Data	Name / description, size, values
T	Polarity tag ("face" / "non-face"), w/ value of 1 or 0
HC	Hash code vector, w/ 9 components
RBM	Reduced binary matrix, 7×7, w/ values of 0 and 1
BM	Binary matrix, 21×21, w/ values of 0 and 1
RM	Relief matrix, 21×21, w/ values of 50, 125 and 200

Based on the observation that certain symmetry exists in the human faces, for accelerating the building / growing of the *knowledge base*, each trained example is mirrored about its vertical symmetry axis. Its mirrored representations are automatically generated and learned with the normal ones, for each example.

It should be emphasized that, before adding any new example, it is checked if wasn't already identically learned in the *knowledge base*, case in which it isn't doubled, avoiding redundancy.

For efficiently computing each *macro-pixel*'s value in the initial *gray levels matrix* (MG) to which each target window is normalized, an *integral image* [21] is computed and used. This one is computed in a single step from the original

image in 256 gray levels, and is a matrix with same dimensions W×H, in which each component gets as value the sum of all of the pixels located to the left and above its corresponding pixel in the original image, inclusively. This way, each *macro-pixel*'s value in MG isn't computed as average of all pixels that it replaces. Instead, it is always obtained at any position and scale through only two additions, a subtraction and a division, by using the corners of the area covered by the respective *macro-pixel* in the *integral image*.

For the cases of possibly multiple candidates obtained at detection for a same face, as mentioned in the previous section, it has been developed a simple algorithm, based on clusterization of overlapped candidates with centroid covered by all the other candidates. For each such cluster, only a representative member is kept, having centroid and sides dimension computed as weighted averages, in which candidates with similarity scores over an adaptive threshold computed per cluster being considered twice. Other candidates partly covered by the respective cluster, if any, are filtered. When clusters of multiple candidates are detected, and also other isolated or smaller candidates, these latter ones are also filtered.

It must be mentioned that our method has a limitation, however reasonable, due to the 21×21 pixels minimum size of the scanning window: faces with lesser resolution (smaller dimension) than this one can't be detected.

4. Experimental system. Results.

An experimental system has been designed and developed for testing the learning mechanisms and the detection algorithm described in the previous sections.

The implementation and development have been realized for Windows® 32-bit, using the C programming language and the Win32 API, compilations being performed with Borland® C++ 5.5, free command line tools.

Experiments were done on an Intel® Core™2 Duo @ 2.66 GHz CPU, with 2 GB RAM, and Windows® XP SP3 operating system.

Simple operations, mainly with integers, were employed. Several programming "tricks" and optimizations have been used. However, it is rather a *proof of concept* implementation.

A unified mechanism for decoding, loading and displaying images from files in BMP, PGM, PNG, GIF, TIFF, PCX, and JPEG formats, with automatic conversion from colors to 256 gray levels when necessary, has been implemented. Image files may be interactively selected, either one by one or several at once for being loaded, displayed and analyzed successively.

On each image, a flexible square window may be interactively defined by clicking and dragging with the mouse, and then adjusted or moved for appropriately framing a human face to be learned in the current *knowledge base*. Alternatively, on a loaded and displayed image, one may start the automatic detection based on the currently loaded *knowledge base*. The detections are graphically marked on the image by framing squares, red colored for all possible candidates, and green colored for the finally determined as representative ones. The current *knowledge base* may be interactively selected (when several available). While learning, an automatic detection on the current image using the *knowledge base* as is at the respective moment may be tried first. This allows to frame for learning as positive examples only faces that were not detected yet, and/or as negative examples only detected *false-positives*.

There have been provided means for the user to externally setup and adjust, for experimental necessities, several parameters as: the step while sliding the scanning window over the image, the type of current learning at a certain moment (positive / negative examples), the similarity thresholds (n_thr, 3_thr, b_thr, r_thr), whether all the detected candidates to be marked or only the representative ones, whether certain filters of the cascade be skipped, whether to use pre-filtering of the candidates, whether to use the tree structure of the *knowledge base*, or whether to shrink the original image before analyzing it.

For learning and testing, there have been used images both from public reference sets (BioID, Vision Group of Essex University, CMU+MIT, and Bao) or synthesized ones (University of Regensburg) downloaded using their links from http://www.facedetection.com/datasets.htm and others randomly got from the Internet, and also personal images.

We started first by employing the whole BioID set of 1,520 single-face images (excepting one,

which contains two faces, as shown in Image 4), with 384×286 pixels. A number of 456 examples of faces have been finally used for learning, from 455 images. For each such face, its image mirrored about its vertical axis of symmetry has been also automatically learned.

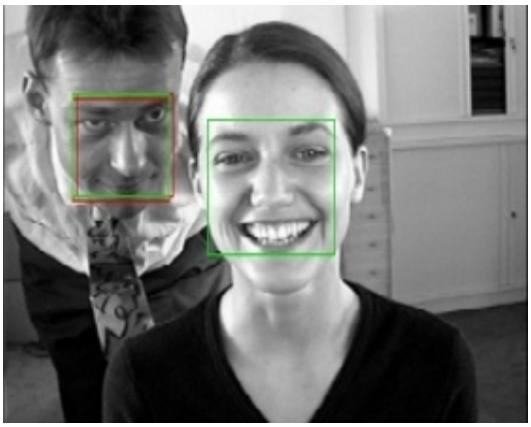

Image 4. Results of detection on a 384×286 pixels image of BioID set (BioID_1140.pgm), without filtering candidates, with scaling factor 1.15:1 and initial step of 3 pixels while sliding the scanning window over image

Detection testing on the other 1,065 single-face images, found almost all of the contained faces, less 14, which represent 1.31% (= 14 / 1,065) *false-negatives*. Most of these 14 faces are either partially truncated at images' borders or masked by fingers, or are too slanted, or their detection was filtered due to some *false-positives* that prevailed. A total of 110 *false-positives* were erroneously detected in certain images from the whole set (one or, at most, two per image), that is 6.80% from all the detections (= *110 / (1,521 – 14 + 110)*). We then also trained as negative examples some of these *false-positives*, together with various other false candidates susceptible to be detected also as *false-positives* (a total of 217 such negative examples, with their mirrored images, too). The result was that the number of undetected faces (*false-negatives*) decreased to 8 that is 0.75% (= 8 / 1,065), with 0 (zero) *false-positives*. This could be finally expressed as a clean detection rate of 99.25% on the test subset, or 99.47% on the whole set.

We continued this interactive guided training, using the single-face sets from Regensburg and parts from Essex (faces94, faces95), and then rechecking backwards for new *false-positives* that were also trained as negative examples. After thus training 809 positive and 708 negative examples, from a total of 4,991 test images, (only) 14 *false-negatives* (0.28%), with 0 (zero) *false-positive*, were detected. This means a detection rate of 99.72%, or 99.76% from all 5,800 faces employed so far.

All in all, we continued our experiments up to a total of 8,998 single-face images, plus a two-faces one. From these ones, 2,000 examples of faces, 1,149 positive and 851 negative, all also being mirrored about their vertical symmetry axis, were used for training. Finally, 261 *false-negatives* (3.32%) and 192 *false-positives* (2.12%) were found while testing the detection on the rest of 7,851 not trained faces, while 7,590 from these ones being detected correctly. That is a detection rate of 96.68%, or 97.10% from all 9,000 faces.

The evolution of the knowledge base and of the detection results is summarized in Table 2.

Table 2. KB training and detection rate evolution

Set(s)	Total Faces	ExP	ExN	FN	FP	FN/ (TF-ExP) %	Det %	DetAll %	FP/ (TF-FN+ FP) %
BioID	1521	375	0	115	80	10.03	89.97	92.44	5.38
BioID	1521	456	0	14	110	1.31	98.69	99.08	6.80
BioID	1521	456	217	8	0	0.75	99.25	99.47	0
+R,E94	3550	598	381	9	37	0.30	99.70	99.75	1.04
+E95	5800	809	708	14	0	0.28	99.72	99.76	0
Stop	9000	1149	851	261	192	3.32	96.68	97.10	2.12

Knowledge base training is still far from being complete, and therefore it is premature and irrelevant to try now a rigorous evaluation, either qualitative or quantitative, for the overall performance of the system. We shall however make a few more comments on the findings of our experiments.

Detection has been tested both with a fixed step of 7, 5 or even 3 pixels added at each scale, and with a scaling factor of 1.10:1, 1.15:1, 1.20:1 and 1.25:1 for scaling the scanning window. Also, while sliding it, there have been tried initial steps of 1, 2 or 3 pixels, increased to ½ of the macro-pixel's side length once the target window grows enough. The number, positions and dimensions of the candidate windows collected at detection depend on these parameters, besides the appropriately adjusted thresholds for various similarity scores.

For images containing relatively big faces (e.g. as in BioID and Essex sets), even initial gliding steps of 3 and 2, and fixed scaling

step of 7 pixels produced acceptable (even good) results at detection. On the other hand, in the case of faces with smaller areas of pixels (e.g. in not so big images with groups of several persons), only an initial gliding step of 1 pixel and a scaling factor of at most 1.10 – 1.15 could provided satisfying results. Obviously, smaller values for these parameters, imply substantially longer computing duration, that could become annoying mainly for big images despite all the optimizations applied. A possible way for diminishing such an impediment could be to automatically reduce enough the image to be analyzed, while still taking care to keep enough resolution on the contained faces, not lesser than 21×21 pixels (as it was the case of the smallest face in the last row from Image 5).

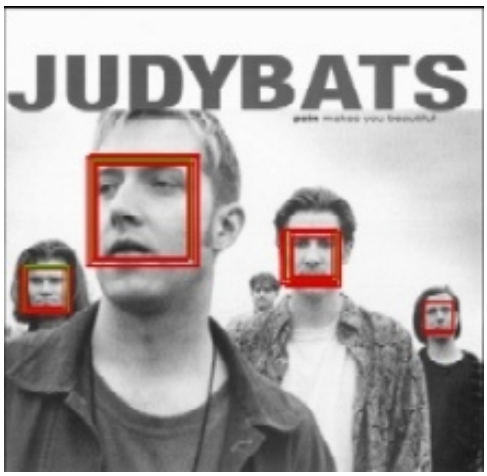

Image 5. Results of detection on an image of the CMU+MIT set (judybats.gif), reduced by 50% from 716×684 pixels, without filtering candidates, with scaling factor 1.15:1 and initial step of 1 pixel while sliding the scanning window over image

The thresholds we used at various comparisons were empirically setup. The *n_thr* threshold for negative examples was set to a higher value, of 70%, for rejecting at the respective stage only those windows that are very similar with learned examples of non-faces. The *3_thr* threshold for the pre-filtering of candidates based on their reduced representations was set to a lower value, of 55%, for not rejecting any window that might somehow contain a face.

The *b_thr* and *r_thr* thresholds used with the main similarity scores were set to a medium value, of 65%, for ensuring an optimal tradeoff between the generalization power and correct discrimination between faces and non-faces.

5. Parallels with other methods

Our proposed method may be compared on portions with the one of *Rowley, Baluja and Kanade* [16] (or, more likely with the one of *Sung and Poggio* [19], partially used and mentioned also in [16]), and, respectively, with the one of *Viola and Jones* [21]. All these methods are based on *supervised learning* of two classes (*face* and *non-face*), from *positive and negative examples*, and then using the gained experience in the automatic face detection process. Similarly, a grayscale input image is analyzed by sliding over it a square window with small dimensions – in our case 21×21, vs. 20×20 or 19×19 and, respectively, 24×24 – in consecutive positions, at various scales. In [16], same as in [19], the scanning of the input image at detection was performed by analyzing a *pyramid of images* obtained by successively scaling the original one. In our case it's the scanning window the one that is scaled and normalized to 21×21 *macro-pixels* at each step and scale. The value of each *macro-pixel* is the average of the covered pixels in the input image, and is computed using an associated *integral image*, as in [21].

The representations that we are using are obtained starting from the gray levels values of the *macro-pixels* in the 21×21 matrix obtained from the scanning window at each step and scale. Simple transformations are applied to become as independent as possible on specific details. This set of representations for each scanning window (HC, RBM, BM, RM) are used by a *pattern matching* algorithm, built as a cascade of consecutive filters with gradual complexity. [19] also uses the gray levels from the scanning window, initially applying some pre-processing for compensating illumination gradient and equalizing the histogram for improving visibility. Then uses a *pattern matching* method, by clustering positive and negative examples, each in 6 representative clusters through a modified *k-means* method and using *k-nearest neighbors*, measuring for each target the distance to the centroids of these distributions at detection. [16] also uses gray levels from the scanning window initially, as [19]. The result is applied as input to a system of multiple neural networks, configured and specialized on certain morphological features specific to component elements of the faces (eyes, nose, mouth). [21] uses a set of *Haar-like features* that are computed based on

the gray levels pixels within rectangular sub-regions of various shapes, dimensions, aspect ratios and orientations within the scanned window. These features are then selectively and successively used in a cascade of composed *Haar filters*.

All these methods are based on learning from numerous positives and negatives examples. All are extremely laborious and intensively computational in the phase of defining, configuring and building their *knowledge base* structures for the classifiers (multiple neuronal in the case of [16], cascade of simple binary filters based on thresholds for the *Haar-like features* configured through an *AdaBoost* meta-algorithm in the case of [21] and, respectively, cascade of simple similarity filters based on thresholds in our case). At detection [21] is the fastest one from all.

All the above-mentioned methods (including our) automatically solve the cases of multiple candidates for a same face, by keeping only one representative candidate, as well as the cases of false candidates, by rejecting them.

We referred here (only) to [16], [19] and [21] considering them (especially [21]) as being among the most significant landmarks for face detection.

Other more recently reported methods [25] are often variations and/or extensions of [21], using Haar-like features or local binary patterns (LBP) or anisotropic Gaussian features filters with AdaBoost type algorithms ([2], [8], [10], [11], [12], [13], [20], [24]), or employ other techniques like support vector machine, SVM ([2], [6], [7], [23]), Haar wavelets ([17]), convolutional neural networks – CNN / ConvNet ([3]), facial landmarks models ([26]), or energy based methods ([14]), while (some) are still using portions of image scanning and pre-processing as in [19] and/or [16]. These methods also demonstrate good (or promising) results, several not only for the frontal-view, but also for the multi-view case.

6. Conclusions

Our proposed approach based on *pattern matching* could be, more or less, comparable until a certain point with *Sung - Poggio* [19]. Anyhow, it differs from this latter one, mainly through: the way the scanning of the original image is performed (by scaling the scanning window as in *Viola-Jones* [21], instead of generating a pyramid of scaled images), the simple averaging with no other pre-processing involved, the multiple gradual representations and the similarity filters applied in cascade on these representations. Also, it differs through the interactive, user guided supervised learning mechanism, implying an initial detection on the current image, followed by the learning of only those yet undetected faces (*false-negatives*) as positive examples, and of the *false-positive* detections as negative examples.

Conducted experiments proved that although this mechanism might appear unwieldy (being anyhow laborious), it is oriented to efficient learning avoiding redundancy in the knowledge base. Also they showed that the pyramid of gradual representations that we proposed (HC, RBM, BM, RM), is quite appropriate to ensure a good generalization and discrimination at the same time due to independence on details while still reflecting the relief of the faces, as well as a good computational efficiency by applying the similarity filters with gradual complexity in cascade (as in [21]).

Even if currently implemented rather as a *proof of concept* and with an incipient, only partially trained *knowledge base*, our proposed method and algorithm seem however to be promising and with potential to be further improved and optimized. These would possibly include: clusterization of the *knowledge base*, keeping only representative members for each cluster, separate comparisons on the upper and lower halves of the faces at detection, automatic reduction of the image dimensions when appropriate, training of an as complete and strong as possible *knowledge base* etc.

Acknowledgements

Presented works were conducted within the frame of the Core National Program TEHSIN, as part of the Project PN0923-0606, at the National Institute for Research and Development in Informatics – ICI, Bucharest, 2013.

REFERENCES

1. Dice, L. (1945). Measures of the Amount of Ecologic Association Between Species. *Ecology,* 26(3), pp.297-302.

2. Faizan, A. Aaima, N., Zeeshan, A. (2012). Image-based Face Detection and Recognition: "State of the art". *International Journal of Computer Science Issues (IJCSI),* 9(6), pp. 169-172.

3. Garcia, C., Delakis, M. (2004). Convolutional face finder: A neural architecture for fast and robust face detection. *IEEE Trans. on PAMI,* 26(11), pp. 1408–1423.

4. Gonzales, R., Woods, R. (2007). *Digital Image Processing. 3rd ed.*, Pearson / Prentice Hall.

5. Hassanien, A. E., Elfattah, M. A., Amin, K. M., Mohamed, S. (2015). A Novel Hybrid Binarization Technique for Images of Historical Arabic Manuscripts. *Studies in Informatics and Control,* 24(3), pp.271-282.

6. Heisele, B., Serre, T., Poggio, T. (2007). A component-based framework for face detection and identification. *International Journal of Computer Vision,* 74(2), pp.167–181.

7. Hotta, K. (2007). View independent face detection based on combination of local and global kernels. In *International Conference on Computer Vision Systems*.

8. Huang, C., Ai, H., Yamashita, T., Lao, S., Kawade, M. (2007). Incremental learning of boosted face detector. *In Proc. of ICCV*.

9. Jaccard, P. (1912). The distribution of the flora of the alpine zone. *New Phytologist* 11, pp.37-50.

10. Jang, J.-S., Kim, J.-H. (2008). Fast and robust face detection using evolutionary pruning. *IEEE Trans. on Evolutionary Computation,* 12(5), pp.562–571.

11. Lin, Y.-Y., Liu, T.-L. (2005). Robust face detection with multiclass boosting. *In Proc. of CVPR*.

12. Mita, T., Kaneko, T., Hori, O. (2005). Joint Haar-like features for face detection. *In Proc. of ICCV*.

13. Meynet, J., Popovici, V. & Thiran, J.-P. (2007). Face detection with boosted gaussian features. *Pattern Recognition,* 40(8), pp.2283–2291.

14. Osadchy, M., LeCun, Y. & Miller, M. (2007). Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research*, pp.1197–1214.

15. Rogers, D., Tanimoto, T. (1960). A Computer Program for Classifying Plants. *Science* 132(3434), pp.1115-1118.

16. Rowley, H., Baluja S., Kanade, T. (1998). Neural network-based face detection. *IEEE Patt. Anal. Mach. Intell. (PAMI),* 20, pp.22-38.

17. Schneiderman, H., Kanade, T. (2004). Object detection using the statistics of parts. *International Journal of Computer Vision,* 56(3), 151–177.

18. Sørensen, T. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analysis of the vegetation on Danish commons. *Kongelige Danske Videnskabernes Selskab,* 5(4), pp.1-34.

19. Sung, K.-K., Poggio, T. (1998). Example-Based Learning for View-Based Human Face Detection. *IEEE Transactions on Patt. Anal. And Mach. Intell. (PAMI),* 20(1), pp. 39-51.

20. Talele, K. T., Kadam, S., Tikare, A. (2012). Efficient Face Detection using Adaboost. *IJCA Proc. on International Conference in Computational Intelligence (ICCIA 2012), ICCIA 10*.

21. Viola, P., Jones, M. (2004). Robust Real-Time Face Detection. *International Journal of Computer Vision (IJCV), 57*(2), Kluwer Academic Publishers, pp.137-154.

22. Vrejoiu, M. H., Hotăran, A. M. (2013). Automatic human faces detection. The Viola-Jones method. *Revista Română de Informatică şi Automatică (in Romanian),* 23(2), pp.21-32.

23. Waring, C. A., Liu, X. (2005). Face detection using spectral histograms and SVMs. *IEEE Trans. on Systems, Man, and Cybernetics – Part B: Cybernetics,* 35(3), pp.467–476.

24. Wu, J., Brubaker, S. C., Mullin, M., Rehg, J. (2008). Fast asymmetric learning for cascade face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 30(3), pp.369–382.

25. Zhang, C., Zhang, Z. (2010). A Survey of Recent Advances in Face Detection. *Microsoft Research Technical Report* MSR-TR-2010-66.

26. Zhu, X., Ramanan, D. (2012). Face detection, pose estimation and landmark localization in the wild. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)* Providence, RI - USA, pp.2879–2886.