# Face Detection Using Combinations of Classifiers

Geovany A. Ramírez Olac Fuentes
Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro No. 1, Tonantzintla, Puebla, 72840, México
geoabi@ccc.inaoep.mx fuentes@inaoep.mx

## Abstract

*In this paper we present a two-stage face detection system. The first stage reduces the search space using two heuristics in cascade: 1) In a face image, the average intensity of the eyes is lower than the intensity of the part between the eyes, and 2) The histograms of the grayscale image of a face with uniform lighting have a distinguishable shape. In the second stage we use combinations of different classifiers including: Naive Bayes (NB), Support Vector Machine (SVM), Voted Perceptron (VP), C4.5 rule induction and Feedforward Artificial Neural Network (ANN); we also propose a simple lighting correction method. We use the BioID face dataset to test our system achieving up to a 95.13% of correct detections.*

## 1. Introduction

Face detection is the first step for applications such as face recognition, computer-human interfaces, and surveillance. Face detection is a difficult task, due to different factors, including varying size, orientation, poses, facial expression, occlusion and lighting conditions [11]. To find faces in images it is necessary to search in the whole image, but frequently the face only covers a small part of the image. We can use very complex methods to scan the whole image, but this can demand a lot of computational resources. One alternative is to use simple methods to search for all the possible faces in the image, and then use complex methods to determine if each possible face is really a face. Sometimes, the face detection process is performed in two or more stages; a coarse detection stage is applied and then followed by a refinement detection. Some systems for face detection are based on the image of the possible face, using the pixel intensity levels as attributes. Other systems measure some features from the image of the possible face to build a set of attributes, and then use these attributes to determine if they correspond to a face.

We present a system to detect faces in gray scale images. The system is divided in two stages; the first is simple, while the second is more complex. The first stage reduces the search space using two heuristics in cascade and improves the lighting of the possible faces. The second stage computes a set of 61 attributes from each image of a possible face accepted by the first stage. Then a combination of classifiers is used to classify the attributes as a face or non-face.

## 2. Related Work

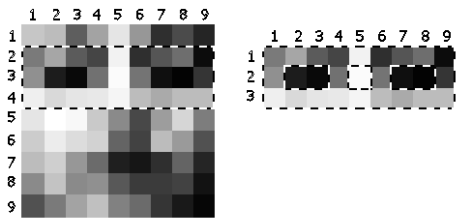
In [8], Rowley et al. developed a face detection system that scanned every possible region and scale of an image using a window of $20 \times 20$ pixels. Each window is preprocessed to correct for varying lighting; after that, a retinally connected neural network is used to process the pixel intensity levels of each window to determine if it contains a face. In [2], Garcia and Delakis used a convolutional neural network. Their system derived automatically convolution filters that act as feature extractors.

In [9], Schneiderman et al. used histograms that represent the wavelet coefficients and the position of the possible face, and then they used a statistical decision rule to determine if the possible face is really a face.

The system of Jesorsky et al. is based on edge images [5]. They used a coarse-to-fine detection using the Hausdorff distance between a hand-drawn model and the edge image of a possible face. In [6], the face model used by Jesorsky et al. was optimized using genetic algorithms, increasing slightly the correct detection rate.

Heisele et al. [4] describe a system based on detection of the face components. They used a two-level hierarchical Support Vector Machine classifier to detect faces. In the first level, the classifier detected the components of the face; in the second level another classifier used the components detected in the first level to match with a geometrical model of the face.

In [10], Viola and Jones used up to 32 steps to detect faces; the first step is a very simple and fast classifier, but with a low accuracy and the last is very complex but with high accuracy.




(a) (b)

**Figure 1. (a) Face region of size $9 \times 9$ pixels and the eyes region (framed with dashed line) corresponds to rows 2, 3 and 4. (b) Eyes Region *I* taken from (a).**

## 3. Proposed System

### 3.1. First Stage

To reduce the search space we look for the possible faces in the image using two simple heuristics. The heuristics are evaluated in cascade on each region of the image. The first heuristic is: *In a face, the average intensity of the eyes is lower than the intensity of the part of the nose that is between the eyes*. We will refer to the heuristic as Eyes-Nose differential (*ENdif*). The whole image is scanned using a sliding window. Each region is resized to $9 \times 9$ pixels using the bilinear interpolation method and after that, histogram equalization is applied, and then the heuristic can be evaluated using Equation 1. Figure 1 shows a face image of $9 \times 9$ pixels and the eyes region *I*, where the heuristic is evaluated. The size of the search window depends on the size of the face to look for; therefore, the window size begins with $30 \times 30$ pixels and is increased iteratively by a factor of 1.15. An image of $384 \times 286$ pixels is scanned in 15 scales. All regions that have an *ENdif* greater than a threshold will be selected to be evaluated using the second heuristic.

$$ENdif = I_{5,2} - \frac{(I_{2,2} + I_{3,2} + I_{7,2} + I_{8,2})}{4} \quad (1)$$

It is not necessary to perform pixel-by-pixel search. Instead, the window can jump an *inc* amount of pixels horizontally, as well as vertically. This increases the speed of the search and reduces the search space. The value of *inc* is calculated with Equation 2.

$$inc = \frac{face\_width}{9} \quad (2)$$

We use the same threshold value used in [7] to select regions of the image that are probably a face.

To improve the lighting of the possible face, an image that complements the lighting of a face image is computed. First, the face image (*face_img*) is divided in 16 regions as shown in Figure 2a; after that, the average value of each region is computed and we use these values to build an adjusted image of $4 \times 4$ pixels. The adjusted image is resized to the size of the face image. After that, each pixel of the adjusted image (*adj_img*) is changed using Equations 3 and 4. Finally, each pixel of the final image (*final_img*) is computed using Equation 5.

$$hv = max(max(adj\_img)) \quad (3)$$

$$adj\_img_{x,y} = hv - adj\_img_{x,y} \quad (4)$$

$$final\_img_{x,y} = face\_img_{x,y} + adj\_img_{x,y} \quad (5)$$

(a) (b) (c)

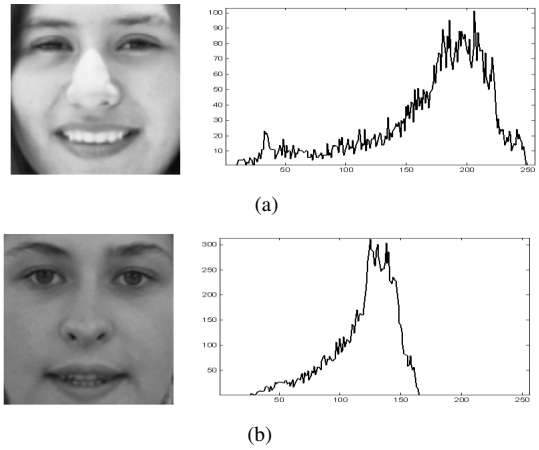
**Figure 2. (a) Face image. (b) Adjusted image. (c) Face image with lighting correction.**

The second heuristic is: *The histograms of the image in grayscale of a face with uniform lighting have a distinguishable shape*. The shape is similar to a Pearson IV distribution. The size and features of the face can modify the appearance of the histogram, but the shape is preserved, as we can see in Figure 3.

Lighting affects the histogram; therefore, first the lighting correction method detailed above is applied. After applying lighting correction to the possible face, we need to normalize the histogram. To perform histogram normalization, the histogram is scaled in magnitude and elements. The magnitude is scaled to the $[-1, 1]$ range and is resized to 35 elements. The peak of the histogram is fixed in position 20. We can see the normalization process in Figure 4.

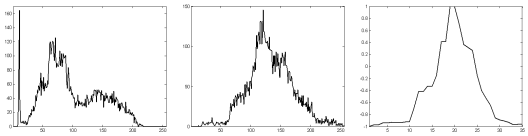
We use a feedforward neural network to distinguish the histogram and a training set of 940 normalized histograms of faces and 940 normalized histograms of non-faces. The network is trained with the backpropagation learning algorithm during 150 epochs.

The first stage can eliminate around 99% of all windows evaluated. Therefore, the second stage will classify only 1% of the possible regions.



(a) (b)

**Figure 3. Examples of the histograms from two face images. The shapes of histograms (a) and (b) are similar, one is wider than the other one and the values are different, this is due to the lighting conditions and the different sizes of the faces in the images.**



(a) (b) (c)

**Figure 4. (a) Original histogram of the face image in Figure 2b. (b) Histogram after light correction. (c) Normalized histogram.**

### 3.2. Second Stage

For each region accepted in the first stage, we compute 61 attributes and then we use 5 classifiers to determine if the possible face is really a face. The classifiers used are: Naive Bayes (NB), Support Vector Machine (SVM), Voted Perceptron (VP), C4.5 rule induction and Feedforward Artificial Neural Network (ANN).

To obtain the attributes we use the binary image of the possible face. The binary image is computed with a threshold function. To calculate the threshold value, we determine the highest value from the histogram of the image with lighting correction, after that, we calculate the cumulative sum from the first element to the element with highest value (see Figure 5). The threshold value corresponds to the index of the cumulative sum from the first element to the element with highest value (see Figure 5). The threshold value corresponds to the index of the cumulative sum that contains 30% of the histogram.

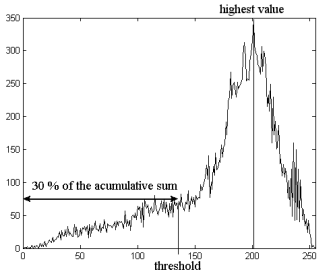



**Figure 5. Histogram from a face with lighting correction. The cumulative sum is from element $1$ to element $201$. The threshold value is $135$.**

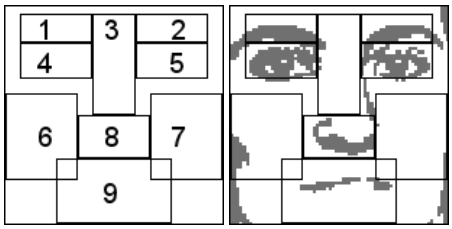
(a) (b) (c) (d)

**Figure 6. Original images with lighting correction (a) and (b). Binary images (c) and (d).**

The binary image will contain the darkest regions in the face, including eyes, eyebrows, mouth and nose. In Figure 6 we can see the binary image calculated with our method.

The binary image is resized to $30 \times 30$ pixels. We calculate the first 8 attributes using a mask with 9 regions, shown in Figure 7. For each region, we calculate its average values using the real value of each pixel instead of the logical value. The 8 attributes correspond to:

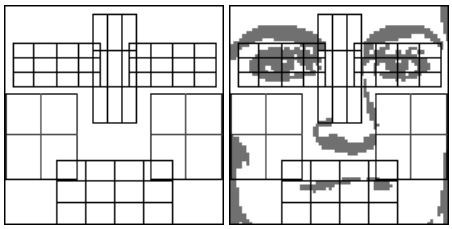
- Difference between regions 1 and 2 (eyebrows).
- Average value in regions 4 and 5 (eyes).
- Difference between regions 4 and 5 (eyes).
- Average value in regions 6 and 7 (cheeks).
- Difference between regions 6 and 7 (cheeks).
- Value of region 3 (nose).
- Value of region 8 (lower nose).
- Value of region 9 (mouth).




(a) (b)

**Figure 7. Mask used to calculate the first 8 attributes.**

The last 53 attributes are calculated using a second mask with 53 regions shown in Figure 8. Each attribute corresponds to the average value of each region. The 61 attributes are normalized to the [-1, 1] interval.



(a) (b)

**Figure 8. Second mask used to calculate the last 53 attributes.**

To build a training dataset, we collected a face dataset of 432 manually cropped images and we added its mirrored images to be a total of 864 face images. To obtain a dataset of non-faces we used the bootstrapping technique as described in [8]. In this technique, instead of collecting the set of non-faces before the training process, the non-face examples are collected during training. In the first iteration the classifiers is trained with the set of faces and a small set of non-faces generated randomly. Then, a set of scenery images that does not contain faces is used to collect subimages to be classified. The subimages misclassified as a face are added to the training set. The classifier is trained again and continues with another iteration. We use an initial non face dataset of 250 images cropped randomly from 10 images that do not present faces. To obtain examples in the bootstrapping process, we use the first stage, and thus we only collect examples that are accepted by the first stage but are not a face.
Each classifier was trained independently with bootstrapping using 25 images that do not present faces during 25 iterations. In the first iteration, the classifier is trained using the original training dataset. The steps for each iteration are:

1. Use the first stage to search for possible faces of a random size.
2. Classify the regions obtained with the first stage.
3. If at least one region was misclassified, add the misclassified examples to the training dataset, otherwise, go to step 1 with another image.
4. Train the classifiers with the updated training dataset.

From each image used to extract examples of non-faces, the first stage evaluated an average of 8300 regions and accepted an average of 70 regions as a possible face. In Table 1 we show the number of regions evaluated for the first stage, remaining images after the first stage and the examples added during all the bootstrapping process for each classifier.

| | **NB** | **SVM** | **VP** | **C4.5** | **ANN** |
|---|---|---|---|---|---|
| regions evaluated | 406206 | 431710 | 523298 | 217096 | 395961 |
| after first stage | 3846 | 4493 | 5674 | 2138 | 4063 |
| examples added | 49 | 62 | 79 | 115 | 106 |

**Table 1. Examples added for each classifier during the bootstrapping process.**

### 3.3. Clustering Overlapping Detections

Frequently a face is detected in different scales, to cluster multiple detections we use the following algorithm:

1. Compute the centroid of each face region.
2. Build clusters of centroids. To build one cluster, first take one centroid and expand its face region by a factor of 1.25. The cluster will contain all the centroids that are found inside of the expanded region.
3. If all the clusters have only one centroid, then the algorithm finishes, otherwise continue with the algorithm.
4. Compute a representative centroid using the average position of all centroids of each cluster, eliminate the remainder and go to step 2.

With the above algorithm we can eliminate some false detections, too. In Figure 9a we can see multiple detections of the face and one false positive, in Figure 9b we can see the final region.

(a) (b)

**Figure 9. Clustering multiple detections.**

## 4. Experimental Results

We use the BioID face dataset [5] to test our system. This dataset consists of 1521 images of $384 \times 288$ pixels, with a frontal view of 23 different persons' faces. This set has a high variety of lighting conditions, background and face sizes. Different tests were performed using single classifiers or combinations of classifiers.

| | NB | SVM | VP | C4.5 | ANN |
|---|---|---|---|---|---|
| Faces detected | 1178 | 1206 | 1366 | 1248 | 1342 |
| Detection rate | 77.45% | 79.29% | **89.81%** | 82.05% | 88.23% |
| False positives | 1171 | 1623 | 1822 | 3639 | 3517 |

**Table 2. Results using single classifier in second stage without cluster.**

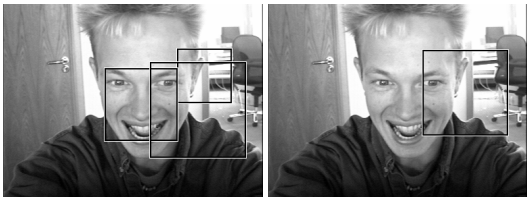
In all tests we use the first stage to reduce the search space. For the first test we use a single classifier in the second stage to determine if the possible face is really a face. We show the result of this test in Table 2 without using the clustering algorithm. Voted Perceptron (VP) reaches the best detection rate of 89.91% with an average of 1.2 false positives per image. In Table 3 we show the result with the clustering algorithm. Again the VP reaches the best rate of 81% with an average of 0.4 false positives per image.

Using the clustering algorithm, the amount of false detection is decreased by up to 60%, but the face detection rate decreases too, because a lot of false positives are near the face and their average position is out of the face region, as we can see in Figure 10.

In the second test we combine the classifiers to improve the detection rate. We test with all the possible combinations of classifiers. If at least one classifier of the combination classifies the regions as a face, then the region is accepted as a face. We show the best result for combinations of 2, 3, 4 and the only combination of 5 classifiers in Table 4 without use the clustering algorithm. With the combination

| | NB | SVM | VP | C4.5 | ANN |
|---|---|---|---|---|---|
| Faces detected | 1114 | 1096 | 1232 | 1022 | 1133 |
| Detection rate | 73.24% | 72.06% | 81.00% | 67.19% | 74.50% |
| False positives | 491 | 606 | 586 | 1157 | 1026 |

**Table 3. Results using single classifier in second stage clustering detections.**



(a) (b)

**Figure 10. Effect of clustering faces and non-faces.**

of all classifiers we reach a rate of 95.13%, but with an average of 4.6 false positives per image. In Table 5 we show the result using the clustering algorithm. The detection rate is worse than that obtained using singles classifiers.

| | 2 classifiers | 3 classifiers | 4 classifiers | 5 classifiers |
|---|---|---|---|---|
| | NB+ ANN | NB+VP+ ANN | NB+VP+ C4.5+ ANN | NB+SVM+ VP+C4.5+ ANN |
| Faces detected | 1390 | 1427 | 1445 | 1447 |
| Detection rate | 91.39% | 93.82% | 95.00% | **95.13%** |
| False positives | 4288 | 5129 | 5240 | 7075 |

**Table 4. Results using combination of 2, 3, 4 and 5 classifiers without cluster.**

In the last test each region is classified and clustered individually by all the classifiers. Then, all regions are clustered at the same time with the same algorithm but the factor used in step 2 is 0.5. The result is shown in table 6. We obtained a detection rate of 93.23% with an average of 1.5 false positives per image.

In Table 7 we compare our results with other systems that use the BioID face data set.

| | 2 classifiers | 3 classifiers | 4 classifiers | 5 classifiers |
|---|---|---|---|---|
| | VP+ ANN | VP+ C4.5 ANN | NB+SVM+ VP+ ANN | NB+SVM+ VP+ C4.5+ ANN |
| Faces detected | 1235 | 1224 | 1179 | 1144 |
| Detection rate | 81.20% | 80.47% | 77.51% | **75.21%** |
| False positives | 790 | 895 | 1222 | 1499 |

**Table 5. Results using combination of 2, 3, 4 and 5 classifiers clustering detections.**

| | NB+SVM+VP+ C4.5+ANN |
|---|---|
| Faces detected | 1418 |
| Detection rate | 93.23% |
| False positives | 2236 |

**Table 6. Result of classify and cluster individually by all classifiers and re-clustered with a factor of 0.5.**

## 5. Conclusions and Future Work

Our system can detect faces under different lighting conditions using 5 classifiers and is slightly more accurate that those reported in [5], [6] and [3]. The first stage can eliminate around 99% of all possible regions of an image. Our method for lighting correction is simple but works very well. The binary image used in the second stage can improve feature detection. Future work will be oriented to reduce the number of false positives.

| | Detection rate | False positives |
|---|---|---|
| Jesorsky et al. [5] | 91.80% | Not reported |
| Kirchberg et al. [6] | 92.80% | Not reported |
| Hamouz et al. [3] | 91.30% | Not reported |
| Fröba and Ernst [1] | 97.75% | 25 |
| our system test 3 | 95.00% | 5240 |
| our system test 5 | 93.23% | 2236 |

**Table 7. Comparison of our system with others.**

## References


[1] B. Fröba and A. Ernst. Face detection with the modified census transform. In *6th International Conference on Automatic Face and Gesture Recognition*, pages 91–96, 2004.

[2] C. Garcia and M. Delakis. A neural architecture for fast and robust face detection. In *IEEE IAPR International Conference on Pattern Recognition*, pages 40–43, Quebec City, 2002.

[3] M. Hamouz, J. Kittler, J.-K. Kamarainen, P. Paalanen, and H. Kälviäinen. Affine-invariant face detection and localization using gmm-based feature detector and enhanced apparence model. In *6th International Conference on Automatic Face and Gesture Recognition*, pages 67–72, 2004.

[4] B. Heisele, T. Serre, M. Pontil, and T. Poggio. Component-based face detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume Vol. 1, pages 657–662, 2001.

[5] O. Jesorsky, K. Kirchberg, and R. W. Frischholz. Robust face detection using the hausdorff distance. In *Third International Conference on Audio- and Video- based Biometric Person Authentication*, Lecture Notes in Computer Science, pages 90–95. Springer, 2001.

[6] K. J. Kirchberg, O. Jesorsky, and R. W. Frischholz. Genectic model optimization for hausdorff distance-based face localization. In *International Workshop on Biometric Authentication*, pages 103–111. Springer, 2002.

[7] G. Ramirez, V. Zanella, and O. Fuentes. Heuristic-based automatic face detection. In *IASTED 6th International Conference on Computer Graphics and Imaging*, pages 267–272, 2003.

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

[9] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 746–751, 2000.

[10] P. Viola and M. Jones. Robust real-time object detection. International Workshop on Statistical and Computational Theories of Vision, 2001.

[11] M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis ans Machine Intelligence*, 24(1):34–58, 2002.