2011 10th Mexican International Conference on Artificial Intelligence

Real Time Face Detection Using Neural Networks

Angel Noe Martinez-Gonzalez and Victor Ayala-Ramirez
Electronics Engineering Department
Universidad de Guanajuato DICIS
Carr. Salamanca-Valle, Km. 3.5+1.8, 36700 Salamanca, Mexico
ayalav@ugto.mx, mtza@laviria.org

***Abstract*—On the one hand, face detection and recognition is an active interdisciplinary area of research that uses techniques from computer vision, image processing and pattern recognition. On the other hand, neural networks have been widely used to address problems in feature extraction, pattern recognition, and in general, the same kind of problems.**

Our proposal here is to use neural networks in the development of a face detection system capable of operating in real time. The system performs a guided face search on interest regions exhibiting human skin color properties. These properties are detected in a pixel by pixel basis. The proposed system can be used as a module of face recognition systems, video surveillance systems, access control systems, for example.

***Keywords*-Face detection; Neural networks; Skin detection; Image processing;**

I. Introduction

Face detection techniques can be classified into two main categories [1].

- Feature-based techniques.
- Image-based techniques.

All these techniques, require *a priori* knowledge of the faces targeted to be detected. Featured-based techniques use explicit face knowledge and the classification is made using information taken from low level features [1]. Typical features used in face detection include: color, face shape and the presence of face elements like the eyes or the nose, for example. Image-based techniques use implicit face knowledge [1], like two dimensional array intensities, and they use a pattern recognition approach to deal with the face detection problem.

As an open area of research, many approaches had been proposed to deal with the problem. The first successful approach of real-time face detection achieving 15 frames per second was proposed by Viola and Jones [2]. Using Adaboost learning, they select the best possible visual features from an *Integral Image* to finally classify it using a combination of classifiers in cascade. Nevertheless, a large number of features is needed along with a long training time. Wang et al. [3] propose a similar idea using also Adaboost learning. They reduce the number of features needed by Wavelet analysis and improve the training time needed by applying a Fisher Linear Discriminant to maximize the ratio between classes. However, the system can not operate int real-time and still needs a long training time. Sahoonizadeh et al. [4] apply PCA to features extracted by Wavelet analysis. The resulting features served as input to a Neural Network classifier which achieves 90.3 percentage of positive detection. The system time process goes from 2 to 4 seconds so it can not operate in real-time. The first successful Neural Network based face detection system was proposed by Rowley et al. [5] which use a retinally connected Neural Network. The system can detect rotated faces and faces with facial expressions in images with complex background and get a low false detection rate. However, the system can not operate in real-time because it has to analyze the whole image. Curran et al. [6] used this idea and reduced the searching space by using a skin guided search in YIQ space color to achieve speed by just looking at the skin regions. The skin regions are processed in gray level intensities to look for faces with a Neural Network classifier. They achieve 67 to 85 percentage of detection rate over a static dataset. Sahoonizadeh et al. [7] use a locally connected Neural Network using only 8 neurons in a single hidden layer as a classifier for Wavelet analysis extracted features. The system achieves efficiency in complex, smiling, partially occluded and lighting variation face images. However the system false detection rate is too high. In the reviewed systems, the structure of the Neural Network structure can be seen as an optimization problem just as the approach of Wiegand et al. [8] who selected the best Neural Network structure by optimizing weights, number of layers and number of neurons needed for a real-time face detection system. Using their optimization algorithm, they improved the detection rate in 50 percent. All the systems reviewed here uses a two dimensional gray level intensities array and considers frontal view faces for the training process.

The system proposed here combines two methods, one based in features and another based on image to detect more accurately faces and to provide a faster response in order to be able to perform in real time.

In our approach, a feature-based method serves as a pre-processing step for the image-based technique. The output of the feature-based method is a set of candidate

978-0-7695-4605-6/11 $26.00 © 2011 IEEE
DOI 10.1109/MICAI.2011.35

regions to contain a face. The main task of this module is to detect skin regions in the image. The image-based method uses a neural network to determine if any part of the candidate region can be detected as a face present in the image. By scanning a face template over the candidate regions, the neural network-based module can detect points where a face can be detected. We use training images of faces and non faces samples in order to determine the neural network parameters.

Main assumptions of the proposed system are:

- Faces exhibit a moderated facial expression, and,
- Faces are shown to the input sensor in a perpendicular pose. i.e. no rotation invariance is considered.

The system proposed here uses an off-the-shelf sensor (a low cost webcam) and its implementation is based in the OpenCV framework available as open source.

II. Skin Detection.

In face detection applications, skin detection is a key component, as shown by the number of works where it has been used [9] [10]. Skin detection is mainly based in verifying if a pixel exhibits a color coordinate that belongs to a specified interval in a given color space.

Skin color modeling can be separated in three main classes:

- Explicit definition of skin models.
- The non parametric modeling of skin.
- The parametric modeling of a skin color distribution.

Our choice for the skin model used in this work was a explicitly defined model. Using this model, the classification rules are simplified and pixel classification can be performed in a faster and more accurate way.

In particular, we have used the normalized RGB color space defined by Equations (1). The color information is only used in the pre-processing space.

$$\begin{aligned} r &= \frac{R}{R+G+B} \\ g &= \frac{G}{R+G+B} \\ b &= \frac{B}{R+G+B} \end{aligned} \tag{1}$$

The use of the normalized RGB color space let us reduce the influence of the illumination variations in the overall performance of our system. We work only with two components of this space, because the third one will be linearly independent of the two other channels. Additionally, the normalized color

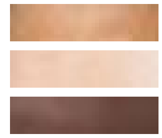


Figure 1. Skin color samples of different skin tonalities used to create the skin detector rules.

coordinates can be obtained using a low computational cost transformation.

The skin model was identified from a set of human skin samples acquired with the same low cost video camera. We have taken the samples from people exhibiting different color tonalities in order to obtain a generic skin color model. Some examples used for identifying the skin color model are shown in Figure 1.

For each pixel in the skin samples, we have computed the normalized *rgb* values. We have then computed first-order statistics for the entire set of skin pixels. We have modeled the skin by using its mean values, μ_r, μ_g and μ_b, the red, green and blue normalized coordinates mean value and its corresponding standard deviation values, σ_r, σ_g and σ_b.

Equation (2) shows the color classification rules used to decide if a pixel in the image can be considered as a skin.

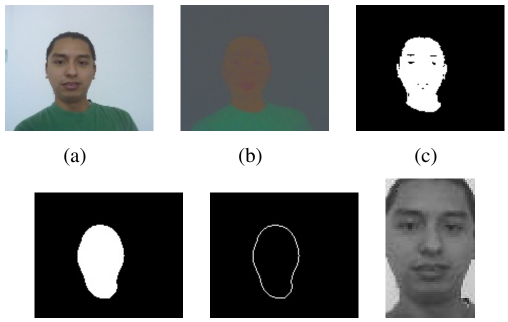
$$\begin{aligned} R_1 &: \ \mu_r - \alpha\sigma_r < r < \mu_r + \alpha\sigma_r \\ R_2 &: \ \mu_g - \beta\sigma_g < g < \mu_g + \beta\sigma_g \\ R_3 &: \ \mu_b - \gamma\sigma_b < b < \mu_b + \gamma\sigma_b \end{aligned} \tag{2}$$

In Equation(2), α, β and γ are channel sensitivity constants that can be tuned for specific conditions in the context of a given application. For example, illumination changes or predominance of a given color in the scene. The larger value for each of these parameters, the larger support of image pixels that will be considered as skin pixels. As the result of the detection step, we get a binary image where the candidate regions are those that have satisfied the skin detection rule. A connected component stage groups pixels into the candidate blobs for face detection. In order to reduce the influence of image noise, a 19×19 median filter is applied.

The candidate regions will be used to locate faces using the neural network-based classifier. A typical result of the skin detection step for an input image is shown in Figure 2.

III. Face Detection.

Neural networks have been widely applied in pattern recognition problems. They can be applied to face detection problems by using as inputs the intensity values of a neighborhood of pixels in the image.



(a) (b) (c)

(d) (e) (f)

Figure 2. Process of skin detection. (a) Input image. (b) *rgb* Normalized transformation. (c) Skin pixels detection. (d) Skin region smoothing. (e) Edge detection of skin regions. (f) Skin portion extracted from the input image.

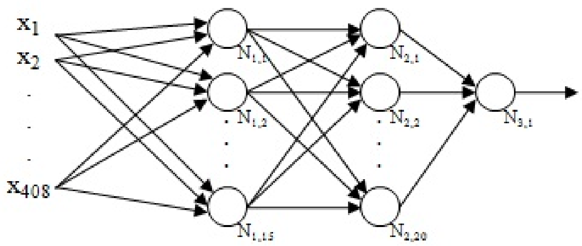


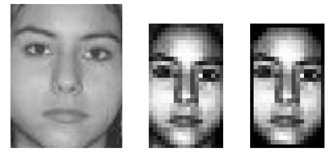
Figure 3. Structure of the three-layer MLP. The inputs correspond to the elliptical gray-level intensity region.

A. Neural Network Setup

The neural network classifier is a three-layer multi-layer perceptron (MLP). The inputs to the NN are 408 gray level intensities taken from an elliptical region fitting in a 30×18 pixel neighborhood (see Figure 3). In Figure 4, we show an example of this pre-processing. Each candidate blob is scaled to satisfy this spatial conditioning.

The first MLP layer consist of 15 neurons having 408 pixels inputs each and where the intensity data is feed. We also use a hidden layer composed of 20 neurons and an output layer with only one neuron. All the neurons use an anti-symmetric activation function, as suggested by Haykin [11]. This choice provides a faster learning time (i.e. it requires less epochs to achieve a particular training error). Layer size has been chosen experimentally to satisfy the compromise between real time performance and accuracy of face detection. Figure 3 shows the structure of the MLP.

The training examples were obtained from the MIT-CBL Face Recognition Database and from the AT&T ORL database. In all cases, color faces were firstly converted into gray level images. A histogram equalization was performed, and finally they were scaled and clipped using the elliptical mask cited above.



(a) (b) (c)

Figure 4. Example of face dataset pre-processing. (a) Gray level skin image. (b) Scaled to 30×18 pixels and equalized image. (c) Filtered image using the elliptical mask.

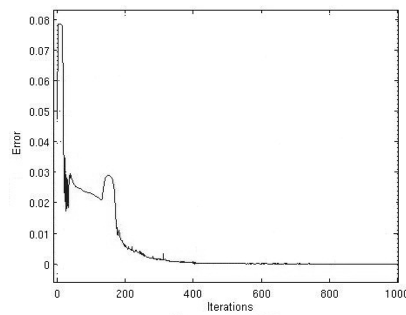


Figure 5. Training error.

The desired outputs of the training samples were labeled as +1.0 if they were a face example and as -1.0 otherwise. The gray level intensity matrix is used as a row vector for the input stage of the neural network.

B. Face Search and Location

When we search for faces in a candidate region, we consider all the positions in the bounding box where a $\frac{1}{2}W \times \frac{1}{2}H$ window can be fitted, with W and H being the width and the height of the candidate region bounding box. For each of these positions, the candidate window is scaled to fit in a 30×18 window and the resulting intensities are presented as inputs to the MLP. The output of the neural network determines if the image region under analysis can be classified as a face instance in the image. If the output is above an experimentally determined threshold of 0.4, a rectangular frame is overlaid on the input image to visually indicate the successful face detection event.

IV. Test and Results

The proposed approach was implemented on a 2.0 GHz 3GB Linux laptop using as input device its built-in camera.

The neural network was trained with 65 patterns of the face class and 95 patterns of the no-face class. The examples presented to the neural network for training were taken from the face detection and recognition databases cited above. Figure 5 shows an outline of the training error through 1000 epochs.

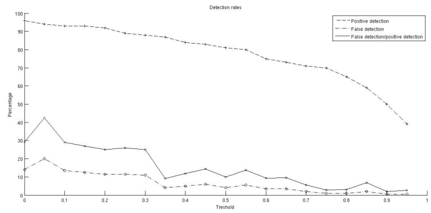


Figure 6. Detection rates for positive detection and false detection.

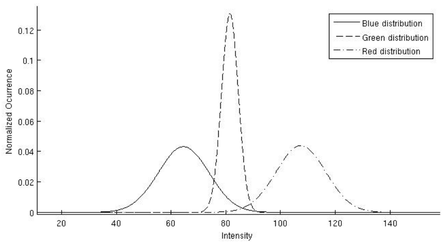


Figure 7. Distribution of the intensities for the skin samples.

We have performed tests both on static images and on real time video acquired by the built in USB camera of the laptop computer.

The output neuron is thresholded to decide if the input presented to the neural network must be classified as a face. We have tested threshold values ranging from 0 to 0.95 in steps of 0.05. We have selected a $\tau = 0.4$ as the threshold setting because the ratio between false detection and positive detection is small enough to give us high accuracy with an acceptable low false detection. Figure 6 represents the positive and false detection rates versus the threshold value. Figure 7 shows the normalized distribution of the intensities for the *r,g,b* channels of the skin samples.

We have chosen to use *r* and *g* channels because the *r* component is associated to the visible color of the skin and the *g* component, because it is the most discriminative color distribution in Figure 7. For the sensitivity parameters, $\alpha = 1.0$ and $\beta = 1.5$ were chosen as the implementation settings.

Figure 8 depicts the steps used for the skin detection step. Figure 8(a) shows the original image. Figure 8(b) shows *rgb* image after normalization. Figure 8(c) presents the result of the raw skin detection process. We perform a median filtering to connect pixels inside the skin blob and to smooth the shape of the candidate regions (Figure 8(d)). The edges of the skin region are shown in Figure 8(e) and Figure 8(f) shows the potential region to be a face instance.

As we can see in Figure 8(c), the skin detector detects false positive regions of skin if the color information of the pixels is very similar to the skin color model. Figure 9 shows the

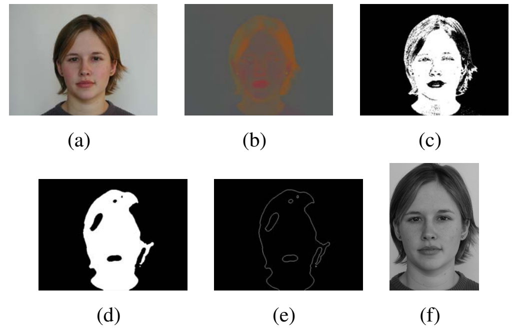


Figure 8. Skin detection processing steps. (a) Input image. (b) Normalized *rgb* transformation. (c) Skin detection result. (d) Median filtering results. (e) Edge detection results over the skin regions. (f) Skin region extracted from the input gray level image.

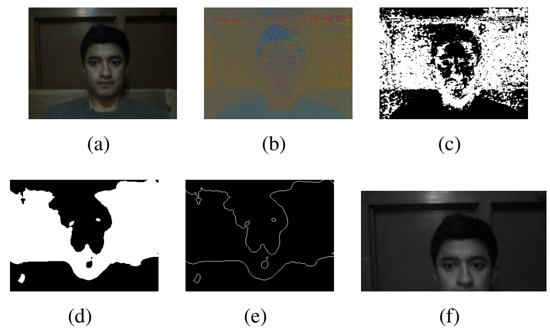


Figure 9. Skin detection on a image with a complex background.(a) Input image. (b) Normalized *rgb* transformation. (c) Background color taken as skin color. (d) Median filter results cover a great part of background. (e) Edge detection results over the skin regions. (f) Skin regions extraction takes a great part of the background as potential face.

results of applying the same procedure to a very similar background (please note skin pixels in white in Figure 9(c)) but it shows also the improvement of smoothing on the results of the skin detection process (see Figure 9(f)).

As we can expect, given the main features of neural network, our system can generalize small variations in the face class. Figure 10 shows an example of the results of an image showing a face with slight variations in rotation and in the facial expression. Figure 11(b) shows also some drawbacks of this generalization because artifacts in the spatial distribution of gray level intensities in conjunction with skin color similarity can provoke errors in the face detection (see the erroneously detected face in the upper right corner).

When the skin detection process fails to output an entire connected region including the face (see Figure 12) the neural network will fail to detect the face. Another case of failure derives when the face region shows a major orientation change with respect to the training patterns (see Figures 13(g) and 13(h)).

(a) (b)

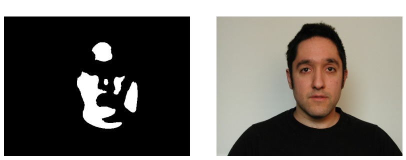
Figure 10. Detection of a face presenting variations in rotation and facial expresion: (a) Skin detection (b) Visual indication.

(a) (b)

Figure 11. A failure situation where a gray level intensity pattern is similar to a face configuration. (a) Skin detection takes great part of background. (b) Face detection considers as a face part of the region of the background filtered as skin.

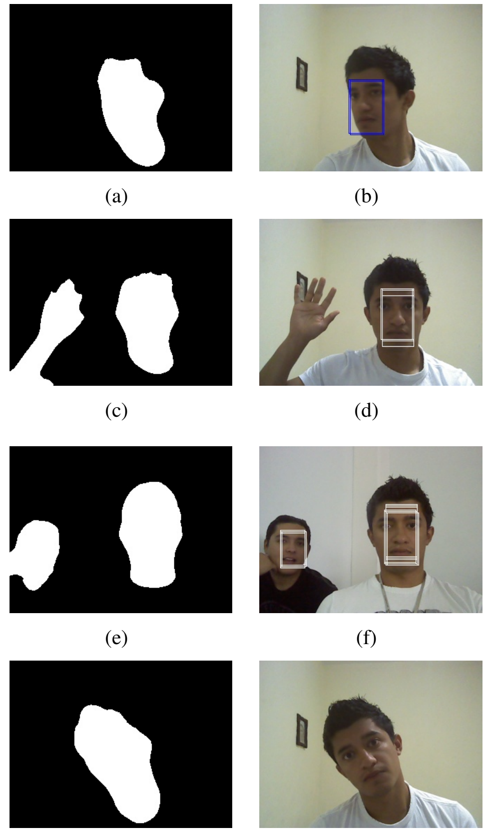
The developed system can also generalize the face detection process when there are multiple instances of faces in the input image as can be seen in Figures 13(c) and 13(d). Finally, Figure 13 presents some frames of the detection process in a real time sequence acquired with the experimental setup described before.

Frame rate performance of the proposed method depends on the area of the potential regions that will be searched. Figure 14 shows the average time ratio of processing one hundred images that exhibit a given area coverage of skin blobs with respect to the whole image area.
We can see that our system can operate with an acceptable frame rate when even if a frame contains between 50 and 70 percentage of the whole area viewed as skin. Nevertheless, if we exceed these values the system goes slow and loses the capability of working in real-time conditions.



(a) (b)

Figure 12. System failure case when the face is separated into several blobs. (a) Detection of unconnected face skin regions. (b) Face detection failure caused by the separated skin regions.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 13. Face detection on a web cam sequence. (a)(c)(e)(g) Skin detection results in real time. (b) Face detection with a little frontal view variation. (d) Face detection discriminating between various skin regions. (f) Face detection results in an image with more than one face. (h) Face detection failure when the face exhibits a rotation variation.

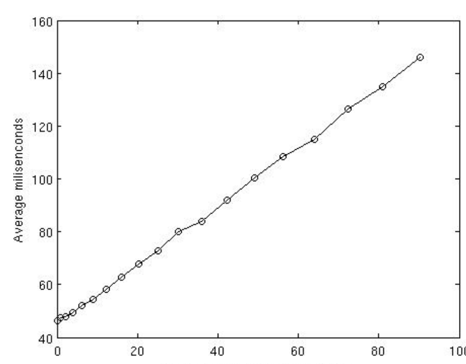


Figure 14. Frame time processing for different area coverage of skin blobs.

V. Discussion

The proposed face detection system works efficiently in real time with a frame rate that goes from 6.6 to 20 frames per second. However, we have identified several issues that we need to improve. Significant illumination variation can lead the system to fail in the skin detection process, because skin does not comply the skin model or because there are disjoint blocks that impede the face detection. We need to explore the use of other color spaces to determine if skin detection can be improved.

Another problem arises is there are objects in the input image that have the same color properties than the skin color model. In such a case, they are considered as potential matching position for the face detection process. This situation makes difficult to satisfy real time constraints because there are more positions to inspect using the neural network face detector.

The proposed system can generalize faces presenting minor variations in facial expressions and face poses. The robustness of the system can be improved by using larger training sets.

References

[1] E. Hjelmas and B. K. Low, "Face detection: A survey," in *Computer Vision and Image Understanding*, vol. 83, no. 3, 2001, pp. 236–274.

[2] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[3] H. Wang, P. Li, and T. Zhang, "Histogram feature-based Fisher linear discriminant for face detection," *Neural Computing and Aplications*, vol. 17, no. 1, pp. 49–58, 2007.

[4] H. Sahoolizadeh, D. Sarikhanimoghadam, and H. Dehghani, "Face detection using gabor wavelets and neural networks," in *Proceedings of World Academy of Science, Engineering and Technology*, vol. 35, November 2008, pp. 553–555.

[5] H. Rowley, S. Baluja, and T. Kanade, "Rotation invariant neural network-based face detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, vol. 20, June 1998, pp. 38–44.

[6] K. Curran, X. Li, and N. McCaughley, "Neural network face detection," *The Imaging Science Journal*, vol. 53, no. 2, pp. 105–115, 2005.

[7] H. Sahoolizadeh and A. Keshavarz, "A FPGA implementation of neural/wavelet face detection system," *Australian Journal of Basic Applied Sciences*, vol. 4, no. 3, pp. 379–388, 2010.

[8] S. Wiegand, C. Igel, and U. Handmann, "Evolutionary multi-objective optimisation of neural networks for face detection," *International Journal of Computational Intelligence and Applications*, vol. 4, no. 3, pp. 237–253, 2004.

[9] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A survey of skin-color modeling and detection methods," *Pattern Recognition*, vol. 40, no. 3, pp. 1106–1122, 2007.

[10] L. Mostafa and S. Abdelazeem, "Face detection based on skin color using neural networks," in *GVIP 05 Conference*, vol. 7, Cairo, Egypt, Dec 2006, pp. 19–21.

[11] S. Haykin, *Neural Networks: A comprehensive Foundation*. New Jersey: Prentice Hall, 1999.