

DETECTING FACES IN COLORED IMAGES USING MULTI-SKIN COLOR MODELS AND NEURAL NETWORK WITH TEXTURE ANALYSIS

Sinan Naji¹, Roziati Zainuddin², Sameem A. Kareem³, Hamid A. Jalab⁴

^{1, 2, 3, 4} Faculty of Computer Science and Information Technology, University of Malaya

50603, Kuala Lumpur, Malaysia

¹sinan@siswa.um.edu.my, ²roziati@um.edu.my, ³sameem@um.edu.my, ⁴hamidjalab@um.edu.my

ABSTRACT

This paper presents an efficient hybrid system to detect frontal faces in colored images regardless of scale, location, illumination, race, number of faces, and complex background. The general architecture of the proposed system encompasses three methods: skin color segmentation, rule-based geometric knowledge, and neural network-based classifier. In the proposed system, multi-skin color clustering models are applied to segment human skin regions, iterative merge stage creates a set of candidate face regions, then, the facial feature segmentation removes false alarms, caused by objects with the color that is similar to skin color. Furthermore, the rule-based geometric knowledge is employed to describe the human face in order to estimate the location of the “face center”. Then the Artificial Neural Network (ANN) face detector is applied only to the regions of the image, which are marked as candidate face regions. The ANN face detector must decide whether a given sub-window of an image contains a face or not. Partial face template is used, instead of the whole face, to reduce face variability, in order to make the training phase easier and to reduce misrecognition. To increase the accuracy of the system, twelve-texture descriptors are calculated and then attached as input data with each face image to train neural network to describe the content of sub-image window such as X-Y-Relieves, smoothness, ratio of darkness, etc. Training neural network is designed to be general with minimum customization. Comparisons with other face detection systems have revealed that our system shows better performance in terms of positive and negative detection rates.

Keywords: *Face Detection, Skin Color Segmentation, Artificial Neural Network, Texture Analysis.*

1.0 INTRODUCTION

Automatic human face detection plays an important role in a wide range of applications such as face recognition, face tracking, human–computer interaction, content-based image retrieval, multimedia applications, and so on. Numerous methods have been developed to detect faces in images such as principle component analysis (PCA) [1] [2], neural network [3] [4], Bayesian classifier [5], template matching [6], support vector machines [7] [8], skin color [9, 10] [11], shape information [10], hidden Markov model [12], fuzzy rule base system [13], Adaboost [14] [15], wavelet packet analysis [16], etc. These methods which have their own advantages and disadvantages are classified into four categories [17]: knowledge-based methods, template matching methods, feature invariant methods, and appearance-based methods. Some methods overlap category boundaries. Extensive survey on detecting faces in still images is presented in [17] [18].

Recently, increasing activities have been noticed in developing approaches to detect faces under unconstrained scene conditions. Among the above-mentioned methods, the appearance-based methods have been gaining more attention. These methods rely on techniques from machine learning and statistical analysis. ANNs have been

successfully applied in many pattern recognition problems. In view of the fact that, face detection can be treated as a two class pattern recognition problem (i.e. face and non-face), various neural network architectures have been proposed [3] [4] and [8, 19]. The face templates are learned from a set of face training images. These learned templates are then used for detection. The main drawback of neural network-based face detector or other appearance-based methods is the huge search space at run time that makes these techniques impractical for real-time systems. Following are some of the main problems [6]:

- Sliding-window technique is widely used to search for faces, but it is very time consuming. This is because the source image is treated without analyzing the contents (i.e. only low-level processing). The sliding window has to be applied at every pixel location in source image, which makes the search space relatively high. As the number of scanning windows in these methods is high, the number of false detects is higher than others methods.
- In order to detect faces at different scales, the source images have to be resized several times (sub-sampling) and the search is repeated again.
- In order to detect faces with varying orientations, the input images have to be rotated several times.

Speeding-up classification stage is a major concern when developing systems for real-world applications. The value of skin-color information has been greatly realized as cue for face detection problem in many face detection approaches [20] [21] [22] [13] [23] [6] [8]. “color has a significant advantage as a surface property and powerful descriptor that often simplifies object identification and extraction from a scene. Moreover, the processing of color information has proven to be much faster than processing of other facial features, which is an important point when dealing with real-time systems”[24].

Generally, dealing with color information involves three main issues such as: (i) selecting the color space, (ii) modeling skin color distribution and (iii) processing of color segmentation results [24]. Several color spaces have been proposed for skin detection. The most widely used color spaces are [13, 25]:

- Basic color spaces (RGB, normalized RGB, CIE-XYZ). [26, 27] [20] [28]
- Perceptual color spaces (HSI, HSV, HSL, TSL) [13, 16, 21] [8] [29] [30].
- Orthogonal color spaces (YCbCr, YIQ, YUV, YES) [31] [32] [33].
- Perceptually uniform color spaces (CIE-Lab and CIE-Luv) [34].
- Others (Mixture spaces, color ratio spaces) [35] [36].

Numerous techniques have been presented in the literature for skin detection using color [25]. Most existing skin segmentation approaches are based on the assumption that the color of skin is quite different from the colors of other objects and its distribution forms a cluster in some specific color space. Skin regions are detected, by identifying the pixels that have skin colors [37]. There are three categories of skin classification and segmentation methods, namely, parametric, nonparametric, and explicit threshold-based skin cluster classifiers [38]. The first category uses parametric models for skin color distributions. This model usually comprises a Gaussian or a mixture of Gaussian models used to model the skin color distribution in different color spaces [1] [39] [40] [41]. The second category uses a non-parametric model. This method estimates the skin color distribution from the histogram of the training data without deriving an explicit model of the skin color [42] [20] [43] [44] [45]. The third category uses explicit rules on color values [46] [35] [47] [16]. This type is generally very simple to implement and is computationally inexpensive. All the above mentioned approaches, explicitly define the boundaries of the skin cluster in a color space. A study of 10 commonly used color spaces for human skin detection and comparisons among them can be found in [48] [49]. A good survey on pixel-based skin color detection techniques was carried out by Vezhnevets et al. [24]. A comprehensive survey on skin-color modeling and detection methods can be found in [25] [22].

The main goal of this work is to show that the face detection problem can be efficiently and accurately solved using a hybrid system that encompasses different methods within its structure to get a cooperative effect between its components, where the strengths of one methodology can compensate the weakness of another. Fig. 1 shows the general system architecture of the proposed system. The proposed hybrid system has the benefit of encompassing image segmentation, geometric knowledge, and neural network-based classifier within its structure. This paper is organized as follows. Section 2, introduces the skin color segmentation. Section 3, presents the rule-based geometric knowledge. ANN classifier is presented in section 4. Sections 5 and 6 provide the experimental results and discussion, Finally, conclusion is presented in section 7.

2.0 SKIN COLOR SEGMENTATION

Most skin-color segmentation techniques involve the classification of individual image pixels into skin and non-skin categories based on pixel color. Algorithms based on skin-color segmentation have to deal with sensitivity to the illumination conditions under which the input image is captured. One of the aspects from which we can observe illumination influence is the choice of proper color space. Color spaces, such as RGB, CYM, CIE, and so on, work very well for technical reproduction purposes. However, human vision tends to look at the world in a different fashion. We do not see the color of an object as quantities of primary mixed colors (red, green, and blue). When humans view a colored object, we tend to describe it by its hue (color). The HSV color space tends to be more realistic in that its representation strongly relates to the human perception of color, thus eliminating the influence of illumination when describing the color of an object. As a result, the HSV model is an ideal tool for developing image processing algorithms based on color descriptions [50] [51] [52]. The transformation between HSV and RGB is nonlinear [44]. However, skin color segmentation is inadequately robust for dealing with complex backgrounds and different image collections. Complex backgrounds usually increase false positive (FP) errors. An FP refers to the proportion of non-skin pixels classified incorrectly as skin. FP can be solved by a system's subsequent steps. Variations in illumination, ethnic groups, and camera characteristics usually increase false negative (FN) errors in which a skin pixel is classified as a non-skin pixel. The FN is most critical problem, attributed to the fact that image segmentation is the first step in image analysis. When a face is missed, the post-processing stages of the system cannot get it back. In this paper, we deal with this problem using reliable, fast, and robust solutions.

Initially, we built a skin-clustering model using the HSV color space and conducted extensive experiments by applying different skin color appearances to identify FN errors. We found two main reasons behind these cases. The first lies on the limitations of the single skin clustering model in covering all skin color appearances, such as dark shadow regions and blackish skin. Strong light reflection may cause skin color information to be lost. In addition, makeup, montage, and image reproduction influence the skin color appearance to a red concentrated appearance.

The second reason lies in the fact that each colored pixel is treated individually in relation to the color space (skin or non-skin pixel), without considering the content of neighboring pixels. Thus, if the skin-color clustering model is too general, it may yield a large number of FP errors, that is, a non-skin pixel classified as a skin pixel, such as furniture, clothes, buildings, rocks, and so on. On the other hand, if the skin-model is tight, then it may yield numerous FN errors, in which the skin pixels are missed.

The skin-color clustering problem leads us to create a novel approach for human skin color segmentation that combines pixel-based segmentation and region-based segmentation in a way that would cover different skin color appearances. Our approach suggests building multi-skin color clustering models, instead of a single skin model, wherein each skin-color clustering model represents the clustering of a pattern class. The skin color models are as follows:

- m_1 : Standard white skin
- m_2 : Shadow skin and blackish skin
- m_3 : Reddish concentrated skin and lips
- m_4 : Light skin

Once we built these clustering models, we segmented the input image into four layers of skin maps, with each layer reflecting its skin-clustering model.

Figure 2 shows segmentation output using multi-skin color clustering models for some test images. Fig. 2(a) shows the output of skin color segmentation at Layer1 for a white-skinned woman with no makeup and under uniform lighting. Fig. 2(b) shows the effectiveness of applying multi-skin color models on images containing people of different races. In this figure, the woman with dark skin is completely recognized at Layer2, although she is misclassified at Layer1. Fig. 2(c) shows an example of various skin tones for different racial groups. The dark reddish hands are correctly classified at Layer3 only. Although the hands that tend to be

yellow are partially classified at Layer1, the convex-hull algorithm can greatly recover the missing regions in such cases. Fig. 2(d) shows that the skin of the same individual varies under non-uniform illumination. The right half of the face with low lighting is correctly classified at Layer2.

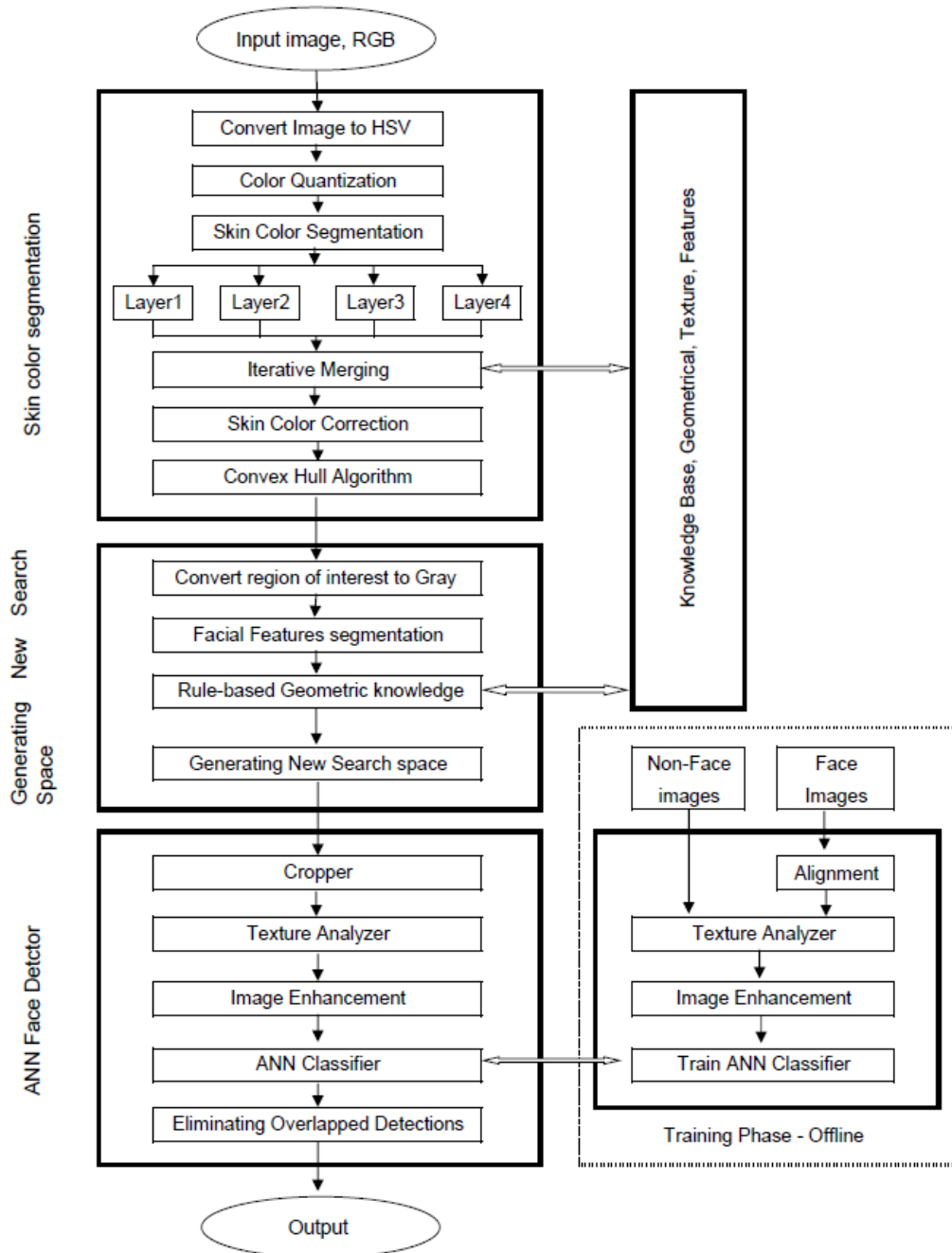


Fig. 1. General System Architecture

The next step in this direction was to use region-based segmentation, in which the analysis of the pixel neighborhood was conducted to perform iterative merging process (region grow) with the other layers. Iterative merge is a procedure that groups pixels or sub-regions into a larger region based on a predefined criterion for

merging. Our merging step started with the objects in the skin layer (Layer1) as the seed points. From these seed points, the region was grown by appending to the seed those neighboring pixels in the other layers that had predefined properties similar to those of the seed.

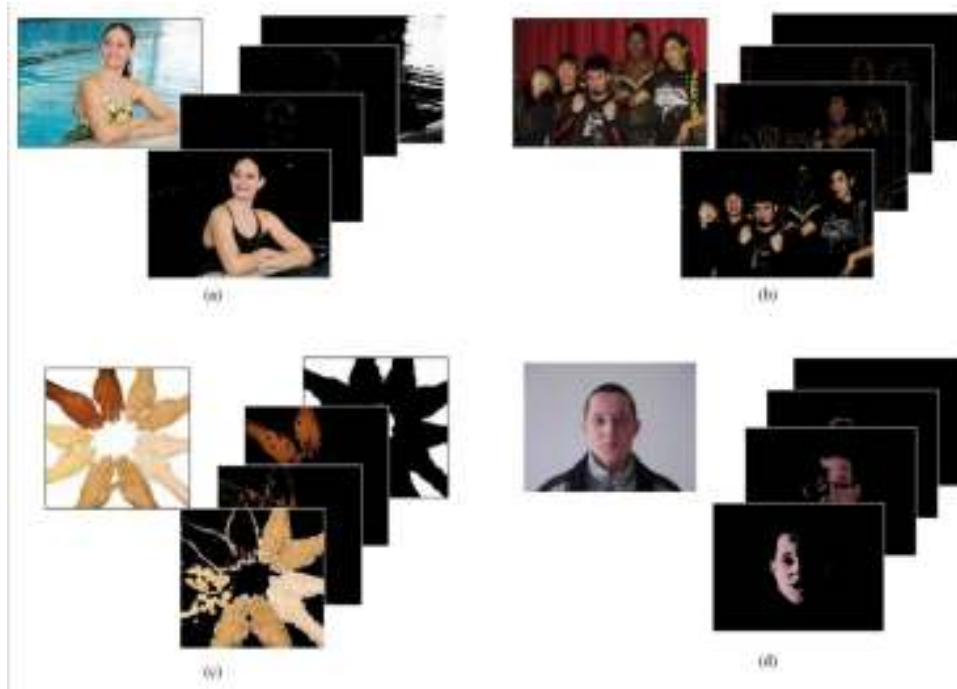


Fig. 2. Human skin segmentation using multi-skin color models; (a) white-skinned woman, no makeup, and uniform lighting; (b) image contains people of different races; the dark-skinned woman is correctly detected at Layer2; (c) various skin tones are detected in different layers; and (d) the skin tone of the same individual varies under non-uniform illumination.

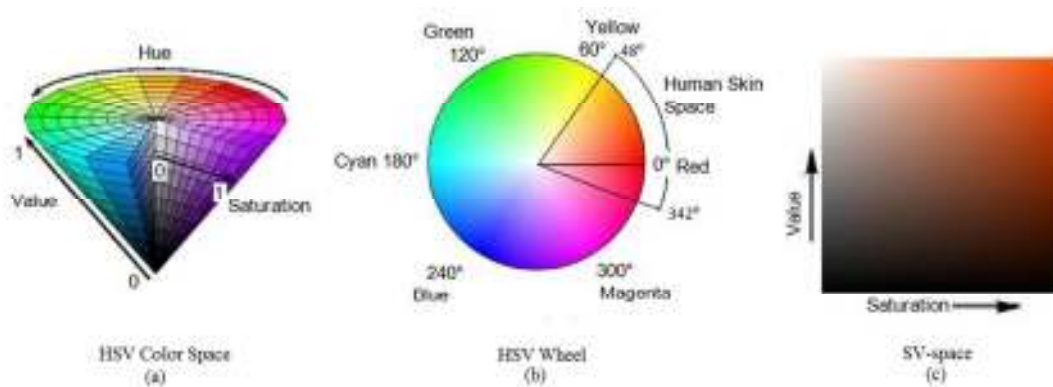


Fig. 3. Converting 3D color space to 2D SV-subspace; (a) HSV color space, (b) HSV wheel shows human skin color space; (c) 2D SV-subspace; where hue = 18.

2.1. Building Multi-Skin Color Clustering Models

Manually, we prepared 2485 patches extracted from still images from different databases for different people and different lighting conditions covering a large range of skin appearances. The skin patches were divided into four pattern classes (m_1 , m_2 , m_3 , and m_4). The skin samples acquired from several regions of human skin, including the forehead, cheeks, nose, neck, and so forth. Skin pixels that were blackish and those containing strong highlights and dark shadows were included. To ensure that skin samples were taken by different cameras, samples were collected from images that came from different sources and types. The non-skin samples are collected from backgrounds manually. The data set is composed of more than 20,000,000 pixels. We found that the skin color clustering for the four skin models was in the range of ($342^\circ \leq \text{Hue} < 48^\circ$) on the HSV wheel as shown in Fig. 3(b). Our purpose is to transform the three-dimensional (3D) color space into a two-dimensional (2D) space without losing color information. We divided the total range of skin color at the Hue channel component into equal intervals on the Hue wheel with ($+6^\circ$). For each quantized constant Hue, the correspondence transformed subspace in the 2D color space shall be as shown in Fig. 3(c). Thus, we have only 12 slides of SV-planes that contain skin color (Hue = 342, 348, 354, 0, 6, 12, 18, ..., 48 degrees). Each colored pixel may now be viewed as a point in a 2D space characterized by the two measurements, saturation S and value V , where ($0 \leq S \leq 100$) and ($0 \leq V \leq 100$).

2.2. Automatic Correction of Non-Uniform Illumination (Lighting Correction)

Non-uniform illumination and varying direction of lighting together with the non-plane shape of the facial features, can cause shadows on the face. Shadows cause shape distortion of objects, false color tone, and loss of facial feature information. However, in real world cases, illumination is not always controllable. Adini [53] observed that, “variations between images of the same face due to illumination are almost always larger than image variations due to change in face identity”. Face detectors, in general, are very sensitive to pixels intensities. For images containing people of different racial groups, the adjustment of darkness level of some skin tones (e.g. African origins) would be very useful in classification stage because it improves the visibility of face details.

Many approaches have been proposed to solve the illumination variation problem for face detection and recognition. Some approaches are global, in the sense that color correction is done by transformation function of an entire image. However, there are cases in which local enhancement is desired. An extensive survey on illumination invariant face recognition can be found in [54].

In this work, we believe that skin segmentation and skin color correction are so closely related that they should not be performed separately. Therefore, illumination correction is done within the iterative merge locally to “regions of interest” rather than the whole image. Local illumination correction can drastically improve the visibility of some facial features as well as correct skin intensity. The goal is to adjust the darkness level of some pixels to provide maximum detail over suitable range of intensities. Then to generate new image as if it has been captured in consistent illumination conditions. Beside, our approach may be used to perform skin color correction on regions in the shadow layer without losing color information. Given that the brightness of the color in the HSV color space is represented in the value component V , we can adjust the overall brightness of region or image by adding a constant bias b to each pixel brightness $g(x,y) = f(x,y) + b$. Similarly, we can adjust the contrast through multiplication of pixels brightness by a constant gain a such that $g(x,y) = a * f(x,y)$. Both methods can be combined to get general equation such that:

$$g(x, y) = a * f(x, y) + b \quad (1)$$

Since, we do not want to specify a constant gain and bias, but would rather map particular range of brightness $[f_1, f_2]$ of dark skin region at layer2 onto a new range $[g_1, g_2]$ of white skin region at layer1, this form of mapping is done using the following equation, adopted from [52] :

$$g(x, y) = g_1 + \left(\frac{g_2 - g_1}{f_2 - f_1} \right) [f(x, y) - f_1] \tag{2}$$

We found that the saturation component S also had effect on the skin color correction of images with highly concentrated red color. Thus, we used the same equation to adjust the saturation component. Fig. 4 shows the idea of skin color correction using multi layers. Fig. 4(a-c) illustrate pixel-based skin segmentation. Fig. 4(d) shows the skin color correction of skin regions at Layer2 by adjusting the darkness of each pixel. The iterative merge used to create a candidate face region is shown in Fig. 4(e). The newly generated image (i.e., lighting-corrected image) is shown in Fig. 4(f).

The drawback of this method is that the visual appearance shows abrupt differences at region's borders after carrying out color correction. Therefore, an average filter is passed over the borders to smooth the region's boundaries.

Both images the original image and the new generated one (i.e. Lighting-corrected image) would be transformed to gray scale and then passed to the ANN face detector. The first row of Fig. 5 shows examples of the source images (gray scale), while the second row shows the lighting-corrected images.

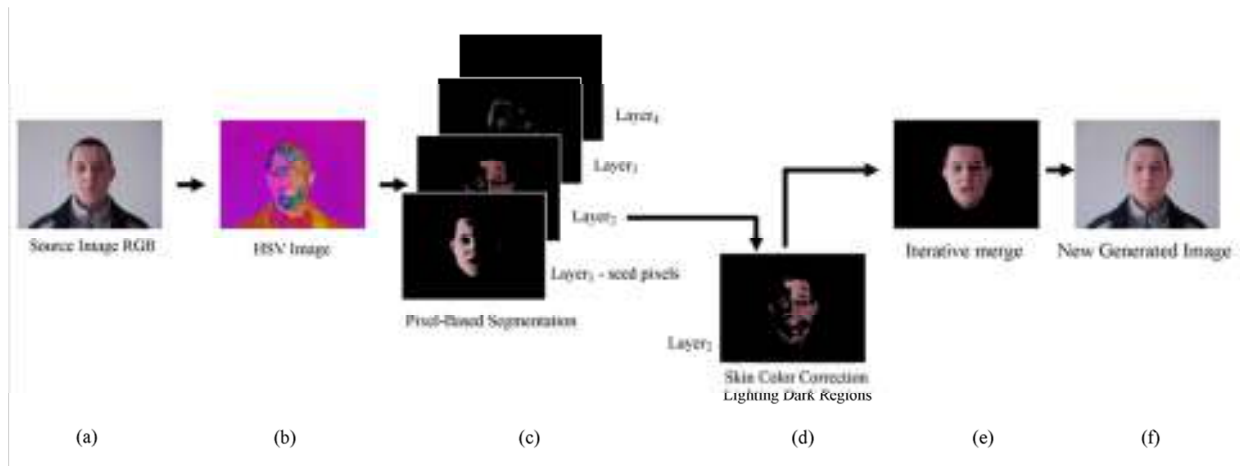


Fig. 4. Human skin segmentation and skin color correction; (a) RGB source image; (b) HSV image; (c) pixel-based image segmentation; (d) skin color correction – lightening dark regions; (e) iterative merge; and (f) newly generated image (i.e. lighting-corrected image).



Fig. 5. Illumination correction. The first row shows examples of the source images (gray scale), while the second row shows the lighting-corrected images.

2.3. Convex-Hull Algorithm

Complex images seldom have very well defined segmented skin-maps. Human skin may be partially detected. Even with a single canonical face, some of the facial features may be lost. In practice, skin segmentation may produce skin-maps that have irregular shapes and may miss some facial features. For example, skin-map that retains only one eye blob which makes face detection harder. This usually occurs when one eye is in touch with other objects such as black hair and consequently it is considered as part of background. In order to retrieve the missing facial features as well as improve the general shape, two image-processing techniques are used:

- i) Morphological closing/opening operations are used for simple border smoothing. The size of the structuring element corresponds to size of the skin-map region. The minimum size is 3-by-3 and maximum size is 9-by-9 square disk-shaped structuring element.
- ii) Convex Hull algorithm is used to approximate the elliptical shape of the face. It is an important step to retrieve misclassified parts of the human face (e.g. missing facial features). Fig. 6(a) shows an output example of skin map that retain only one eye blob. In this figure, the right eye is considered background. Fig. 6(b) shows the usage of Convex Hull algorithm to approximate the elliptical shape of the face. Masking convex hull with source image (gray scale) is used to retrieve the missing facial features (i.e. right eye) as shown in Fig. 6(c).

Generally, convex hull algorithm may attach non-skin region along with skin regions. Next sections describe our solution to overcome this problem.

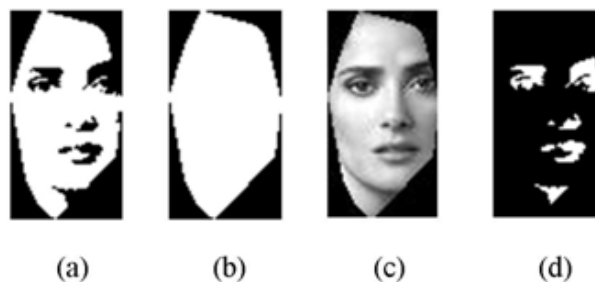


Fig. 6. Convex hull algorithm (a) skin-map retains only one eye blob; (b) Convex hull algorithm is applied to approximate the elliptical shape of the face in order to retrieve the missing eye; (c) masking convex hull with the source image (gray scale); (d) threshold-based segmentation is used to extract facial features.

3.0 RULE-BASED GEOMETRIC KNOWLEDGE

Images with complex backgrounds contain surfaces and objects with skin-like colors. While the main purpose of previous stage is to find the regions in an input image that might potentially contain faces based on skin color, this stage aims at removing false alarms caused by objects with color similar to skin color such as background and other exposed parts of the human body. The final goal of this stage is to locate the candidate “face center” region based on the 2D-geometrical information of face constituents.

“The existence of eyes, eyebrows, nose tip, and mouth blobs evidence that the potential skin segment is indeed a face. Facial feature segmentation is based on the observation that facial features differ from the rest of the face because of their low brightness” [47]. Therefore, this heuristic is used to binarize the skin segments. Initially, skin-maps will be masked with the original image (i.e. gray scale), then apply thresholding. “Thresholding is a fundamental approach in image segmentation that enjoys a significant degree of popularity, especially in applications where speed is an important factor” [50]. The intensity value $f(x,y)$ is regarded a facial feature as follows;

$$g(x,y) = \begin{cases} 0 & \text{if } f(x,y) > T \\ 1 & \text{if } f(x,y) \leq T \end{cases} \quad (3)$$

where T obtained via Otsu's method [55].

Figure 6(d) shows the output of threshold-based image segmentation used for facial features segmentation. Then additional information about each facial feature such as position, shape, area, bounding box, major diagonal, secondary diagonal, etc. are calculated and kept in a separate data structure.

Next, we have to consider the 2D-geometrical information that encodes human knowledge of what constitutes a typical face. Although faces can be very variable, but still have the same basic structure and content. The two eyes are the most distinguishing features for face detection. The two eyes and one nose tip (or mouth) in the frontal view constitute an inverted triangle. Potential face regions are discovered by finding such triangles adopted from [56] [57]. Face triangles are obtained by finding the combination of two eyes and nose tip/ mouth blobs. It is easy to come up with simple rules to describe the face structure that forms an inverted triangle. We have reported here some of these rules:

above(x,y): if object(x) is above object(y).
 left(x,y): if object(x) is to the left of object(y).
 two_eyes(x,y): if left(x,y) \wedge similar(x,y) \wedge distance_match(x,y).
 face(x,y,z): if two_eyes(x,y) \wedge above(x,z) \wedge above(y,z) \wedge distance(x,z,d) \wedge distance(y,z,d)

Many other geometrical measures such as the relative location, shape, and size are considered in our rules. The search starts with primitive facial feature blobs and successively applying the rules. The geometric knowledge is used to guide the process until a total description of a face is obtained. Usually, there are many combinations, which may satisfy the sample pattern. A 2D-list of size $N \times 3$ is generated as shown in Fig. 7(b). Each row in the list corresponds to a face candidate structure consisting of three objects: left eye, right eye and nose tip. Usually, the combinations in the list are overlapped (e.g. $\langle A,B,C \rangle$, $\langle A,B,D \rangle$, $\langle A,B,E \rangle$). Based on simple heuristic that faces rarely overlap in images, many overlapped detections are eliminated to only one as shown in Fig. 7(c). Fig. 7(d) illustrates that each row in the list forms a “triangle” that corresponds to a potential face.

The rectangular area lying between each eye pair is considered as a potential “face center” region (see Fig. 7(e)). This area is projected into a binary image as a set of pixels of value 1, whereas the rest of the image is set to 0 as shown in Fig. 7(f). The figure shows two areas with pixel value of 1 that will be regarded a candidate face center instead of entire face region. The new generated binary image would be our new search space, instead of the whole image. In other words, many skin regions of the image would be discarded early, while spending more computation on promising face-center regions.

Our experiments reveal that some small faces are missed because some geometric rules did not hold. Therefore, if the size of the face candidate region is so small (e.g. area ≤ 500 pixels such that 25×20 pixels), there is no need to apply geometrical rules (i.e. the entire region is passed directly to next step). This reduces the

probability of missing small faces. Experimental results on complex images show that human skin detection and geometrical information are not enough to detect human face, as it needs other processing steps.

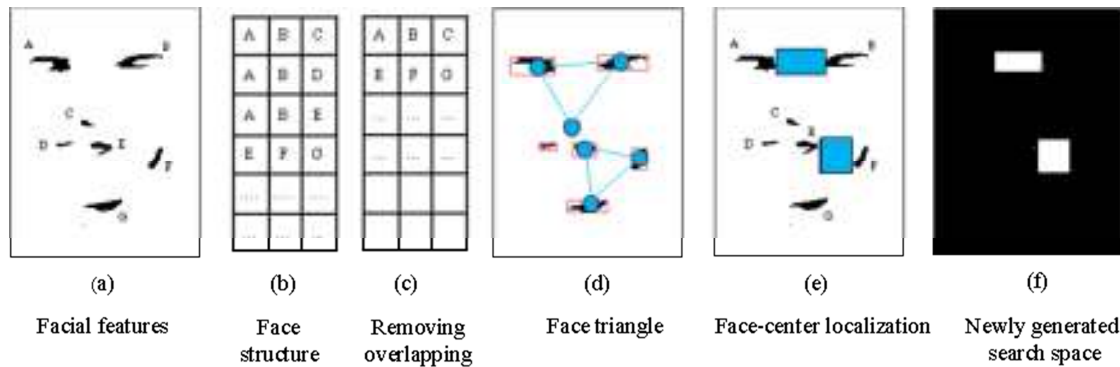


Fig. 7. Face-center localization; (a) an example of facial features; (b) list of combinations that successfully describe face candidate structure; (c) removing overlapped detections; (d) corresponding potential faces displayed as inverted triangle; (e) face-center localization; (f) binary image representing the newly generated search space.

4.0 ARTIFICIAL NEURAL NETWORK CLASSIFIER

The last part of the proposed system performs the task of verifying the detection of a face (i.e. classification). We have proposed an efficient face detector based on Artificial Neural Network (ANN). The ANN face detector decides whether a given window of an image contains a face or not.

Due to the positive impact of the pioneer approach of Rowley et al. [3], many other researchers had followed this approach [16] [58] [59] [60]. We have proposed to use texture descriptors with partial face pattern and to improve the face detection architecture of Rowley et al. [3]. The proposed classifier is preliminarily presented by [61] to detect faces in gray scale images. More details, findings, and comparisons are presented hereby.

4.1. Partial Face Pattern

The first step to improve the accuracy of the classifier is to reduce the amount of variation between images of training faces. This would give the most compact space of images of faces.

As far as our knowledge, all previous architectures have used whole face pattern to train the classifier. One of the challenging problems in the training task is the high variability in the training faces due to many issues such as facial expressions, illumination, presence or absence of structural components (e.g. beard, mustaches and glasses), etc. It is clear that reducing the variability will be helpful. Standard deviation (σ) is used to figure out the most variable facial features for the same individual and then for different individuals. It is calculated as in [2] :

Step 1: Obtain face images I_1, I_2, \dots, I_m (training faces) where m is the number of faces.

The face images must be *centered* and of the same size.

Step 2: Represent every image I_i as a vector Γ_i

Step 3: Compute the average face vector Ψ :

$$\psi = \frac{1}{m} \sum_{i=1}^m \Gamma_i \tag{4}$$

Step 4: Compute the standard deviation vector (σ):

$$\sigma = \frac{1}{m} \sqrt{\sum_{i=1}^m (\Gamma_i - \psi) (\Gamma_i - \psi)^t} \tag{5}$$

The standard deviation vector (σ) shows that the lower part of the face has higher variations (i.e. the presence or absence of structural components such as beard and mustache combined with high degree of deformability of mouth) which makes training harder. In this work, the partial face pattern is used instead of whole face, to reduce face's variability. Our face pattern consists of eyebrows, eyes, cheeks, down to nose tip as shown in Fig. 8. The size of face pattern is fixed experimentally to 15×23 pixels. Excluding the lower part of the face makes the training easier, reduces the size of data, and improves the detection rate. As the number of positive detections increased, the numbers of false detections will increase correspondingly.

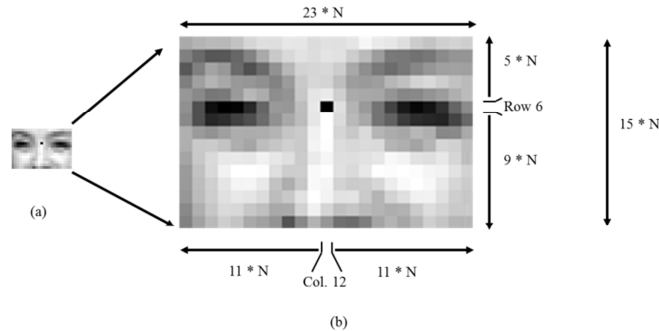


Fig. 8. Partial face pattern; (a) of size 15×23 pixels with face center at location (6, 12); (b) the ratio of distances in relation to the face center where N is variable to capture larger faces.



Fig. 9. X-Y-Relieves: (a) Y-relief; (b) X-relief

4.2. Face Center Labelling and Alignment

The second step in reducing the amount of variation between images of faces is to deal with alignment sub-problem; i.e. aligning all training faces so that the approximate positions of the facial features are the same in all images. This alignment should reduce the variation in the two-dimensional position, orientation, and scale of the training faces.

Many researchers have carried out this task manually. Rowley et. al [3] handled this problem by labeling many points in each training face (i.e. eyes, tip of nose, and corners of the mouth). Then, these points were used to normalize each training face to the same scale, orientation, and position. From our point of view this normalization causes distortion to some faces, due to the changes caused by the geometric transformations, since faces differ from one to another. In this work, aligning of faces was done using only one point. We refer to this point as “face center”. It is defined as the intersection of two lines: the first one passes horizontally through both eyes and the other vertically upwards from the tip of the nose. The position of each facial feature relative to that point is considered in preparing our training faces. The location of the face center would be at pixel (6,12). Fig. 8 shows the geometrical information of facial features relative to the face center. The usage of distance ratios makes our approach relatively insensitive to the size of the face.

4.3. Texture Analysis

The main complexity in training neural network is exacerbated by the fact that we are dealing with huge chunks of data. Consider our 15×23 pixels face pattern; we have 256^{345} possible combinations of gray values. It is extremely high dimensional space. Even with a lot of training data, there is a probability of misclassification. The main reason lies in the fact that each pixel's intensity, x_i , represents the i^{th} descriptor in the feature vector without any consideration to the spatial relationships between them.

In this work, the feature vector is richer, in which not only the intensities of pixels are considered but also the content of neighboring pixels (i.e. spatial relationship between them to describe the content of face image). It is assumed that, human faces have a distinct texture and the same type of facial features will have the same brightness (i.e. cheeks). The face pattern is divided into five regions, namely: two eyes, two cheeks, and nose tip. Therefore, a set of texture descriptors are measured from these regions and attached with the image as input data to form a new feature vector. An important approach for describing a region is to quantify its content using statistical properties. Mean (m), standard deviation (σ) and smoothness (r), adapted from [62], are measured for predefined regions (i.e. eyes, cheeks, nose forehead and nose tip). Smoothness measures the relative smoothness of the intensity in a region. These descriptors are defined as follows:

$$m = \sum_{i=0}^{L-1} z_i P(z_i) \tag{6}$$

$$\sigma = \sqrt{\sum_{i=0}^{L-1} (z_i - m)^2 P(z_i)} \tag{7}$$

$$r = 1 - 1/(1 + \sigma^2) \tag{8}$$

where z_i is a random variable indicating intensity, $p(z)$ is the histogram of the intensity levels, and L is the number of possible intensity levels [62].

Then X-Y-Relieves, adapted from [47] [63] are measured for each training face by processing the horizontal and vertical profiles. The Y-relief is obtained by summing all pixel intensities in each row. Similarly, X-Relief is obtained by summing all pixel intensities in each column. The horizontal (row) and vertical projections are defined as follows:

$$HI(x) = \sum_{y=1}^n f(x, y), \quad VI(y) = \sum_{x=1}^m f(x, y) \tag{9}$$

Then, it is easy to locate facial features by detecting the local maximum (or minimum) and first abrupt transition. As shown in Fig. 9(a-b), each facial feature generates a maximum in Y-Relief and has specific X-Relief characteristics.

Throughout the texture descriptors extraction stage, twelve descriptors are measured. These are:

- M₁: smoothness of the left cheek region.
- M₂: smoothness the right cheek region.
- M₃: local minimum, Y-Relief for eyes.
- M₄: local minimum, Y-Relief for nose tip.
- M₅: local maximum, X-Relief for nose forehead.
- M₆: darkness ratio of left eye darkness to left cheek.
- M₇: darkness ratio of right eye darkness to right cheek.
- M₈: darkness ratio of left cheek darkness to right cheek.
- M₉: darkness ratio of left eye darkness to nose forehead.
- M₁₀: darkness ratio of right eye darkness to nose forehead.
- M₁₁: darkness ratio of left cheek darkness to nose tip.
- M₁₂: darkness ratio of right cheek darkness to nose tip.

By using the ratios of darkness between selected facial features, the method becomes relatively insensitive to the illumination, being able to cater for pixel intensities and the darkness ratios between various features for relative feature intensities in different illuminations. The texture descriptors, M_i , are entered with each image as input data to form richer feature vector for training NN.

4.1. Training Neural Network

Generally, training phase needs a lot of face and non-face samples. A classifier is trained using standard multilayer back propagation neural network from a database of 17,200 image samples. There are 5,200 positive samples of face patterns and the rest are non-face. The training phase involves two stages: extracting texture descriptors, and training the classifier to learn the feature vector. Our face images are prepared as follows:

- 1) Faces are cropped manually from images. Usually, source images contain faces of various sizes, orientations, positions, and intensities. All training faces are frontal faces rotated up to $\pm 10^\circ$.
- 2) “Face center” is labeled manually for each face (i.e. it is defined as the intersection of two lines: the first one passes horizontally through both eyes and the other vertically upwards from the tip of the nose) as shown in Fig. 10(a-b).
- 3) The system starts to crop many sub-images at different window sizes but with the same face center. The system increases the window size repeatedly by ratio of 1.1 and crop a new sub-image window at that size for each step; leading to a pyramid of face images. This will guarantee that the location of face center would appear at same predetermined location in all faces, see Fig. 10(c). Then, move the face center one pixel in all directions (i.e. 8-neighbors pixels) and repeat step 3 for eight iterations.
- 4) Images that meet certain level of similarity are selected manually for next steps. Images that do not resemble faces are excluded.
- 5) All images are resized to 15×23 pixels as shown in Fig. 10(d).
- 6) In order to reduce variability due to lighting conditions and camera gains; histogram equalization is applied to improve contrast.
- 7) Texture descriptors of each image are measured and attached to image as input data.

The next step is to prepare “non-face” patterns. This is considered as one of the challenging tasks in training neural network. It is hard to get a representative sample of non-face images. The problem is how to describe and characterize “non-face” pattern. We can say that any sub-image that does not contain a face can be a characterized as a non-face image. This makes the space of non-face images very large compared to the face images. Instead of collecting the non-face images by hand, non-face images are collected as follows:

- 1) A set of source images are selected at random.
- 2) Demolish all faces to obtain images which contain no faces.
- 3) Run the system to crop 10,000 sub-images at random window sizes and locations.
- 4) Resize sub-images to 15×23 pixels.
- 5) Apply the preprocessing step to improve contrast and normalize intensities.
- 6) Images that resemble faces are discarded.
- 7) Texture descriptors of each image are measured and attached to each face image as input data for training ANN.

During training stages, false detections are added to the training database as new non-face examples. About 2000 non-face examples are added in this way. Fig. 11 shows examples of face and non-face images used for training neural network classifier.

The collected images are divided into three subsets: training, validation, and testing. The training set is used first to train the ANN, the validation set is used to further refine the ANN architecture. The test set is used to measure the performance of the ANN.

There are 357 neurons (345 pixels + 12 texture descriptors) at the input layer; each one represents one feature of our feature vector. Ten neurons in the hidden layer 1, five neurons in the hidden layer 2 and one neuron in the output layer, which generates an output ranging from -1 to +1, signifying the presence or absence of a face, respectively. The particular choices of number of neurons per layer were driven empirically through a trial and error process in which the number of neurons was increased until a significant reduction in the false positive rate could be achieved.

The network’s weights are initialized with random values. Then, face and non-face images are repeatedly presented as input with the corresponding desired targets. The output is compared with the desired target, followed by error measurement and weights adjustment until the correct output for every input is reached. The

hidden layers neurons are estimated using activation functions that feature the hyperbolic tangent sigmoid transfer function, whereas, the output layer neuron is estimated using the activation function that features the linear transfer function.

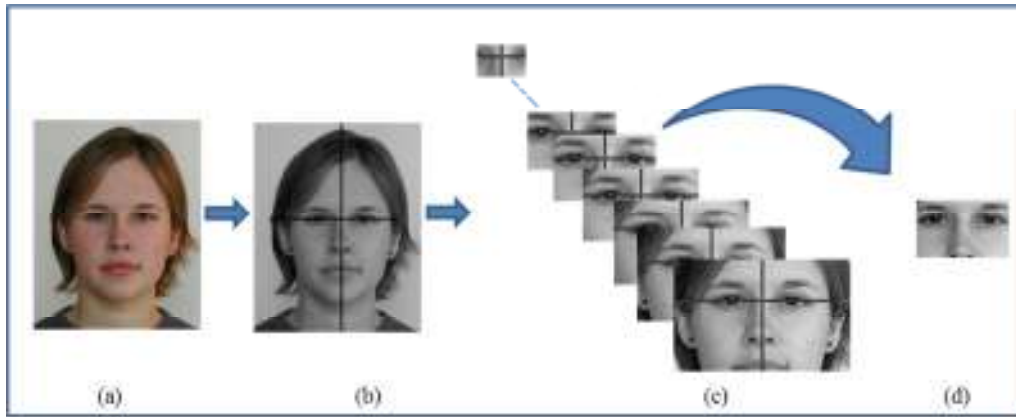


Fig. 10. Preparing training faces (a) source image. (b) “Face center” is labeled manually (c) pyramid of face images with the same face center (d) selected training face, resize it to 15×23 pixels.



Fig. 11. Examples of face and non-face samples used for training neural network face detector.

4.5. ANN Classification Phase

The ANN classification phase consists of four steps: the cropper, histogram equalizer, texture-analyzer and ANN face detector. The cropper receives as input three images of the same size: the source image (i.e. gray scale), a binary image identifying search space, and the lighting-corrected image as shown in Fig. 12(a). The function of the cropper is to crop a sub-image window from the source image, and passes it to the next step. To detect faces with different size, the cropper increases the window size repeatedly in the same manner as in training phase leading to a pyramid of sub-images as shown in Fig. 12(b). To detect faces anywhere in the input image, the cropper moves to every location with value of 1 in the search space and starts cropping. Each cropped sub-image is resized to 15×23 pixels. Histogram equalizer improves contrast as shown in Fig. 12(c-d). The texture-analyzer is used to measure twelve texture descriptors from each sub-image and attached them as input data to be passed to the ANN face detector.

The face detector functions as follows: it receives as input a 15×23 pixels sub-image plus twelve texture descriptors. It decides whether the sub-image contains a face or not. If the sub-image is classified as a face, then its location is saved as shown in Fig. 12(g). Nevertheless, if it is classified as non-face, the classifier checks the corresponding sub-image at the lighting-corrected image. Therefore, there is no need to perform lighting correction for each sub-image at classification time. This step is highly important because it speeds up the system.

To speed up the classifier, the cropper uses a fixed size window as a sliding window of 15×23 pixels at the search space. To detect faces with different sizes, the system repeatedly reduces the source image by 0.1 and the cropper starts cropping at that size.

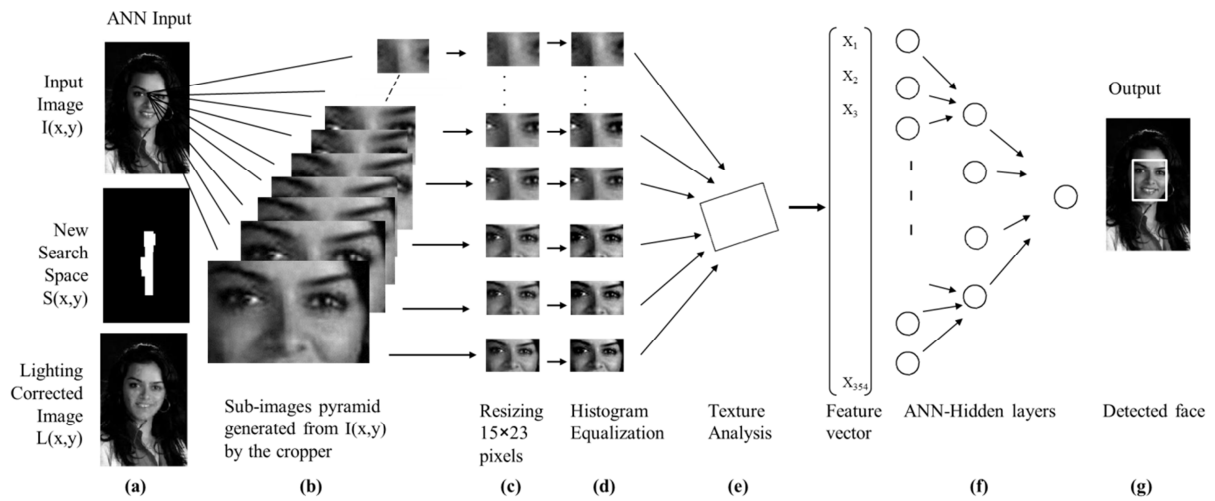


Fig. 12. ANN-based face detector; (a) The ANN input; (b) sub-images pyramid; (c) resizing to 15×23 pixels; (d) Histogram equalization; (e) texture analysis; (f) the hidden layers of neural network classifier; (g) the output image.

4.6. Eliminating Overlapped Detections

Generally, the ANN face detector may produce multiple positive detections for a face, because the same face can be detected at multiple scales and at several nearby positions. This leads to multiple overlapped detections. It is clear that, eliminating these overlapped detections is one of the functions of the system. The location of face centers of these detections tend to cluster about a typical nearby region. Since faces rarely overlap in images, we have proposed to eliminate such detections using a familiar approach of pattern matching based on measures of distance between pattern vectors. The system computes a set of Euclidean distances ($d_i(x)$) between each member of detections and our predefined sample face pattern (i.e. $d_1(x), d_2(x), \dots, d_n(x)$). If the i^{th} positive detection is identified as the best detection that resembles our pattern, then it yields the smallest Euclidean distance:

$$d_i(x) < d_j(x) \quad j=1,2,\dots,n; j \neq i \tag{10}$$

In other words, an unknown detection is said to be the best detection if its sub-image is most similar to our pattern. The system keeps the location of such detection and eliminates the other detections. Fig. 13(a-c) shows an example of eliminating overlapped detections.

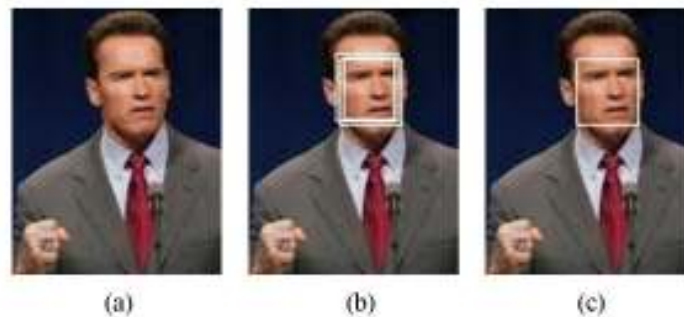


Fig. 13. Eliminating overlapped detections: (a) source image, (b) multiple overlapped detections and (c) eliminating overlapped detections to only one.

5.0 EXPERIMENTAL RESULTS

The experimental results are presented to show the effectiveness of the proposed system. Our system was carried out in Matlab 2010a on Intel(R) Core i5 at 2.2GHz, 4GB DDR3 Memory, system type 64-bit, Window 7. To evaluate the proposed system, four different databases were used for training and testing:

- The “CVL Face Database” of the Computer Vision Laboratory in University of Ljubljana [64]. This database includes 114×7 images for each person. Each image has the size of 640×480 pixels. Images were taken under different lighting directions, expressions, and poses.
- The “LFW database including FDDB” [65] contains more than 13,000 face images collected from different resources .
- The “JAFPE database” [66] The Japanese Female Facial Expression Database contains 213 images of 7 facial expressions (gray scale 256×256 pixels). This database is used for preparing part of training faces to train ANN classifier.
- We have created our face database images (FSKTM database). This database consists of 540 randomly collected images from the Internet. In general, Internet images are acquired under unknown conditions. These images, therefore, have been taken by different cameras and of course under various lighting conditions.

We have chosen sample images from the above-mentioned databases containing 245 faces of different size, illumination, location and complex background. When a “face” is detected in source image, the system draws an appropriate bounding box at the corresponding face. The bounding box is drawn larger than our pattern size to be corresponding to actual face size (i.e. for clarity). Fig. 14 shows some face-detection results; (a) using CVL dataset; (b) using FSKTM dataset; (c) using LFW dataset. As there are single canonical face images with uniform background in CVL dataset, the detection rate is very good (i.e. 97.14%). Considering the unconstrained nature of internet images containing many faces using FSKTM and LFW, the detection rate is slightly lower, but it is still good (i.e. 92.47% and 89.02% respectively). Fig. 15 shows two positive detections and one false face detection (i.e. sub-image window that resembles the face pattern). The figure also illustrates that there are many natural non-face objects/patterns in the real world, which are look like face pattern when considered in separation. Since we use small patterns for classification, the existence of such patterns in the image poses a problem to the face detection systems. Three sets of experiments were performed to evaluate system performance: A conventional image-based feature vector, texture-based feature vector, and integrated image and texture feature vector. The comparison among performance evaluations is shown in table 1. The table shows the effectiveness of the proposed face detector, which is based on integrating both image and texture features. The execution time required to detect faces in the source images depends the size, image content colors, and complexity of images.



Fig. 14. Examples of detected faces; (a) using CVL dataset; (b) using FSKTM dataset; (c) using LFW dataset.



Fig. 15. Two positive detections and one false detection (i.e. sub-image window that resembles the face pattern).

Table 1. Performance of ANN-based face detector using different feature vectors

Type of Feature Vector	Positive Detections		False Detections	
	Set A	Set B	Set A	Set B
Image-based feature vector	89.80%	83.60%	9.20%	12.30%
Texture-based feature vector	94.40%	84.50%	7.30%	8.20%
Integrating image and texture feature vector	95.30%	86.50%	2.40%	4.10%

6.0 DISCUSSION

The comparative analysis of our face detection system with other systems is shown in table 2. The table shows the best results of different methods. Although different methods use different face databases in training and testing, the table gives an overall picture about the performance of the methods. Our system can detect between 86.5 – 95.3 percent of faces for a set of 205 images containing 245 faces, revealing that the proposed system, relatively has the lowest false detections than other systems.

Zaqout *et al.* [20] have developed a two phased face detection system in colored images. First, it applies skin color segmentation to detect skin-like regions based on lookup-tables in RGB color space. In the second stage, the system detects faces based on the assumption that the appearance of face is blob-like and has an approximately elliptical shape. The main differences as follows:

- Their system used RGB color space, which is not perceptual. This makes image segmentation harder because of its limitations as shown before.
- Their system based on assumption that skin segmentation produces skin maps with approximately elliptical shape. However if the source images contain surfaces and objects with skin-like colors or other exposed parts of the human body such as hands, shoulders, legs, etc. the system shows high false negatives and false positives.

However, in our work the system does not depend on preconditions and assumptions. In practice, we found that skin detection is not enough to detect human faces and other processing steps are needed. Our approach resolves false alarms caused by skin-like regions through subsequent stages to compensate the weakness of skin color segmentation.

Table 2. Performance of our ANN-based face detector compared to other face detection methods.

Method	No. of faces	Detected faces	Detection rate	False detection rate
Skin color + LS Face Model [9]	725	680	93.8%	NA
Skin color + PCA [1]	486	470	96.7%	7.4%
Neural Networks [3]	507	390 - 461	76.9 - 91.1%	~ 14.41%
Bayes Classifier [67]	441	124	85.5%	20.63%
Color Clustering [68]	118	83	70.3%	6.6%
Fuzzy HSCC + SVM [69]	NA	NA	93.6%	16.3%
Skin color + Wavelet Packet[16]	104	98	94.23%	19.23%
Skin color + Ellipse fitting [20]	NA	NA	~94.17%	~17.31%
Support Vector Machine [7]	155 + 313	115 + 304	74.2 - 97.1%	0.1 – 12.9%
Our System CVL dataset	70	68	97.14%	2.85%
FSKTM dataset	93	86	92.47%	4.30%
LFW dataset	82	73	89.02%	4.87%

Our classifier is originally inspired by the well-known pioneer face detection approach developed by Rowley [3], but has the following valuable improvements:

- Rowley’s approach is originally designed to detect faces from gray scale images. We have developed this approach to be applied for color images.
- The search space of Rowley’s approach is relatively high as other appearance-based face detectors. Nevertheless, the sliding-window technique is widely used but it is very time consuming. The classifier examines every location in the input image. However, in this work, the search space is restricted to regions of interest while excluding the background.

- Color information is a powerful descriptor that often simplifies object identification and extraction. The reduction in the number of sliding windows would consequently reduce false detections due to background.
- To improve the detection rate, our system uses partial face template instead of whole face. This minimizes misrecognition caused by mustache, beard, and high degree of deformability of mouth.
- Rowley's approach uses pixel intensities of the face image as descriptors to train neural network. In this work, additional twelve texture descriptors are computed for each face image, and then attached as input data with each face image to train neural network. This will enhance the reliability of the system, but at the expense of more computations. When detection methods are used within systems, it is important to consider requirements, speed and accuracy. Accuracy may need to be sacrificed for speed or vice versa.
- Our system uses only one point, i.e. the "face center", to align all training faces. This keeps the training faces as they are, whereas Rowley's approach uses many points. These points were used to normalize each training face to the same scale, orientation, and position. This normalization causes distortion to some faces.
- Rowley's approach performs lighting correction for each sliding window at classification run time. If the search space is big, the system scans more windows and thus the computational cost is high. As reported in Ref. [3], a linear function is used to approximate the overall brightness of each part of the sub-image window and then to be subtracted from it to compensate for a variety of lighting conditions as shown in Fig. 16. In our approach, lighting correction is done prior to classification stage and the resultant image is saved along with the source image. Our classifier checks the lighting-corrected image only when the source sub-image fails (i.e. non-face). This increases the search process up to double, but it is still better than performing the lighting correction procedure for each sliding window at classification run time.

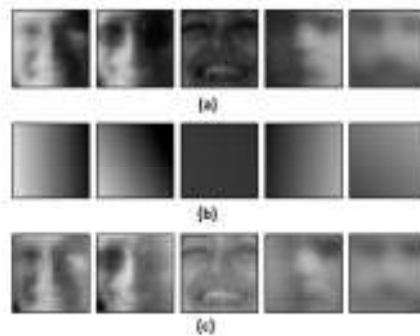


Fig. 16. Non-uniform lighting correction of Rowley's approach (a) Original sub-image (b) best fit linear function (c) lighting corrected sub-image [3].

7.0 CONCLUSION

In this paper, we had presented a novel hybrid system to frontal detect human faces from colored images. The proposed system has high accuracy to detect faces regardless of size, position, expression, illumination, number of faces, race, and complex background. A novel multi-skin models based on the classification rules contain information about the skin colors of various ethnic origins and illuminations was used. The proposed hybrid system has the benefit of encompassing three methods within its structure: image segmentation, geometric knowledge, and neural network. With such architecture, we have a cooperative effect among its components, where the strengths of one method can compensate for the weakness of another. The experimental segmentation results of multi-skin color models revealed that, it not only improved the segmentation accuracy but also allowed us to exploit more information about skin regions and the relationship between regions, which is very important issue in image segmentation. The contribution of multi-skin color models is motivated by the idea that skin segmentation and skin color correction are so closely related that they should not be performed separately.

It is noteworthy that the proposed partial face pattern can be used as an efficient way to reduce face variability, which consequently results in enhancing the detection rate, because it reduces misrecognition (i.e. false negatives) caused by mustache, beard, and high degree of deformability of mouth. There are many natural non-face objects/patterns in the real world, which are look like face pattern when considered in separation. Since we use small patterns for classification, the existence of such patterns in the image poses a problem to the face

detection systems. These cases showed the difficulty of building an accurate face detection system without detecting the other surrounding known objects such as hair, neck, shoulders, etc. to verify the existence of human face.

REFERENCES

- [1] Shih, F.Y., et al., *Extracting faces and facial features from color images*. International Journal of Pattern Recognition and Artificial Intelligence, 2008. 22(3): p. 515-534.
- [2] Turk, M.A. and A. Pentland, *Eigenfaces for recognition*. Journal of cognitive neuroscience, 1991. 3(1): p. 71-86.
- [3] Rowley, H.A., S. Baluja, and T. Kanade, *Neural network-based face detection*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 1998. 20(1): p. 23-38.
- [4] Feraund, R., et al., *A fast and accurate face detector based on neural networks*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2001. 23(1): p. 42-53.
- [5] Schneiderman, H. and T. Kanade, *A statistical method for 3D object detection applied to faces and cars*. Proceedings. IEEE Conference on Computer Vision and Pattern Recognition, 2000: p. 1746.
- [6] Jin, Z., et al., *Face detection using template matching and skin-color information*. Neurocomputing, 2007. 70(4-6): p. 794-800.
- [7] Osuna, E., R. Freund, and F. Girosi, *Training support vector machines: an application to face detection*, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1997: San Juan, 1997. p. 130.
- [8] Juang, C.F. and S.J. Shiu, *Using self-organizing fuzzy network with support vector learning for face detection in color images*. Neurocomputing, 2008. 71(16-18): p. 3409-3420.
- [9] Hiremath, P. and A. Danti, *Detection of multiple faces in an image using skin color information and lines-of-separability face model*. International Journal of Pattern Recognition and Artificial Intelligence, 2006. 20(1): p. 39-62.
- [10] Wang, J. and T. Tan, *A new face detection method based on shape information*. Pattern Recognition Letters, 2000. 21(6-7): p. 463-471.
- [11] Zainuddin , R. and S. Naji. *Multi-skin color clustering models for face detection*. in *Second International Conference on Digital Image Processing, Proceedings of SPIE*. 2010. Singapore.
- [12] Nefian, A. and M.H. Hayes III. *Face detection and recognition using hidden Markov models*. in *International Conference on Image Processing*. 1998. IEEE.
- [13] Moallem, P., B.S. Mousavi, and S.A. Monadjemi, *A novel fuzzy rule base system for pose independent faces detection*. Applied Soft Computing, 2011. 11(2011): p. 1801-1810.
- [14] Viola, P. and M. Jones, *Robust real-time face detection*. International Journal of Computer Vision, 2004. 57(2): p. 137-154.
- [15] Wu, B., et al., *Fast rotation invariant multi-view face detection based on real adaboost*. Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004: p. 79-84.
- [16] Garcia, C. and G. Tziritas, *Face detection using quantized skin color regions merging and wavelet packet analysis*. Multimedia, IEEE Transactions on, 1999. 1(3): p. 264-277.

- [17] Yang, M.H., D.J. Kriegman, and N. Ahuja, *Detecting faces in images: A survey*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2002. 24(1): p. 34-58.
- [18] Hjeltnæs, E. and B.K. Low, *Face detection: A survey*. Computer Vision and Image Understanding, 2001. 83(3): p. 236-274.
- [19] Lin , S.H., S.Y. Kung, and L.J. Lin, *Face recognition/detection by probabilistic decision-based neural network*. Neural Networks, IEEE Transactions on, 1997. 8(1): p. 114-132.
- [20] Zaqout , I., R. Zainuddin, and S. Baba, *Human Face Detection In Color Images*. Advances in Complex Systems, 2004. Vol. 7(3): p. 369–383.
- [21] Terrillon, J.C., M. David, and S. Akamatsu, *Detection of human faces in complex scene images by use of a skin color model and of invariant Fourier-Mellin moments*, in *Fourteenth International Conference on Pattern Recognition* 1998, IEEE Computer Brisbane, Qld., Australia. p. 1350-1355
- [22] Schmugge, S.J., et al., *Objective evaluation of approaches of skin detection using ROC analysis*. Computer Vision and Image Understanding, 2007. 108(1-2): p. 41-51.
- [23] Li, B., X. Xue, and J. Fan, *A robust incremental learning framework for accurate skin region segmentation in color images*. Pattern Recognition, 2007. 40(12): p. 3621-3632.
- [24] Vezhnevets, V., V. Sazonov, and A. Andreeva, *A survey on pixel-based skin color detection techniques*. Graphicon 2003. 3: p. 85-92.
- [25] Kakumanu, P., S. Makrogiannis, and N. Bourbakis, *A survey of skin-color modeling and detection methods*. Pattern Recognition, 2007. 40(3): p. 1106-1122.
- [26] Eveno, N., A. Caplier, and P.Y. Coulon, *New color transformation for lips segmentation*. IEEE Fourth Workshop on Multimedia Signal Processing 2001: p. 3-8.
- [27] Dargham, J.A. and A. Chekima, *Lips detection in the normalised RGB colour scheme*. Information and Communication Technologies ICTTA 2006. 1: p. 1546-1551.
- [28] Chen, H.Y., C.L. Huang, and C.M. Fu, *Hybrid-boost learning for multi-pose face detection and facial expression recognition*. Pattern Recognition, 2008. 41(3): p. 1173-1185.
- [29] Zainuddin, R., S. Naji, and J. Al-Jaafar, *Suppressing False Negatives in Skin Segmentation*, in *Lecture Notes in Computer Science LNCS*, T. Kim, et al., Editors. 2010. p. 136-144.
- [30] Sandeep, K. and A. Rajagopalan. *Human face detection in cluttered color images using skin color and edge information*. in *Indian Conference on Computer Vision, Graphics & Image Processing*. 2002. ICVGIP 2002, Ahmadabad, India.
- [31] Hsu, R.L., M. Abdel-Mottaleb, and A.K. Jain, *Face detection in color images*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2002. 24(5): p. 696-706.
- [32] Kumar, C. and A. Bindu, *An efficient skin illumination compensation model for efficient face detection*. 32nd Annual Conference on IEEE Industrial Electronics, 2006: p. 3444-3449.
- [33] Saber, E. and A.M. Tekalp, *Frontal-view face detection and facial feature extraction using color, shape and symmetry based cost functions*. Pattern Recognition Letters, 1998. 19(8): p. 669-680.
- [34] Cai, J. and A. Goshtasby, *Detecting human faces in color images*. Image and Vision Computing, 1999. 18(1): p. 63-75.
- [35] Chen , W.C. and M.S. Wang, *Region-based and content adaptive skin detection in color images*. International journal of pattern recognition and artificial intelligence, 2007. 21(5): p. 831.

- [36] Tan, W., et al., *A Fusion Approach for Efficient Human Skin Detection*. Industrial Informatics, IEEE Transactions on, 2012(99): p. 1-1.
- [37] Phung, S.L., A. Bouzerdoum, and D. Chai. *Skin segmentation using color and edge information*. in *Seventh International Symposium on Signal Processing and its Applications, 2003*. . 2003. Paris, France.: IEEE.
- [38] Abdullah-Al-Wadud, M., M. Shoyaib, and O. Chae, *A skin detection approach based on color distance map*. EURASIP Journal on Advances in Signal Processing, 2008. 2008: p. 199.
- [39] Liu, Z., J. Yang, and N.S. Peng, *An efficient face segmentation algorithm based on binary partition tree*. Signal Processing: Image Communication, 2005. 20(4): p. 295-314.
- [40] Caetano, T.S., S.D. Olabarriaga, and D.A.C. Barone. *Performance evaluation of single and multiple-gaussian models for skin color modeling*. in *Symposium on Computer Graphics and Image 2002*. IEEE.
- [41] Kim, C., et al., *Color segmentation robust to brightness variations by using B-spline curve modeling*. Pattern Recognition, 2008. 41(1): p. 22-37.
- [42] Gomez, G., M. Sanchez, and L. Enrique Sucar, *On selecting an appropriate colour space for skin detection*. MICAI 2002: Advances in Artificial Intelligence, 2002. Volume 2313: p. 3-18.
- [43] Ma, Z. and A. Leijon. *Human skin color detection in RGB space with Bayesian estimation of beta mixture models*. in *18th European Signal Processing Conference (EUSIPCO-2010)*. 2010. Aalborg, Denmark.
- [44] Phung, S.L., A. Bouzerdoum, and D. Chai, *Skin segmentation using color pixel classification: analysis and comparison*. IEEE transactions on pattern analysis and machine intelligence, 2005: p. 148-154.
- [45] Chai, D., S. Phung, and A. Bouzerdoum, *A Bayesian skin/non-skin color classifier using non-parametric density estimation*. Proceedings of the International Symposium on Circuits and Systems, 2003. 2: p. II-464-II-467.
- [46] Kovac, J., P. Peer, and F. Solina, *Human skin colour clustering for face detection*. The IEEE Region 8 EUROCON 2003 Computer as a Tool 2003. 2: p. 144-148
- [47] Baskan, S., M.M. Bulut, and V. Atalay, *Projection based method for segmentation of human face and its evaluation*. Pattern Recognition Letters, 2002. 23(14): p. 1623-1629.
- [48] Chaves-González, J.M., et al., *Detecting skin in face recognition systems: A colour spaces study*. Digital Signal Processing, 2010. 20(3): p. 806-823.
- [49] Terrillon, J.C., et al. *Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images*. in *Fourth IEEE International Conference on Automatic Face and Gesture Recognition, 2000*. . 2000. Grenoble, France.: IEEE.
- [50] Russ, J.C., *The image processing handbook*. Fifth ed. 2007: CRC Press, Taylor & Francis Group.
- [51] Gonzalez, R.C., R.E. Woods, and S.L. Eddins, *Digital Image Processing Using MATLAB*. 2007: Prentice Hall Press.
- [52] Efford, N., *Digital Image Processing: A Practical Introduction Using Java* 2000: Addison-Wesley Longman Publishing Co., Inc.
- [53] Adini, Y., Y. Moses, and S. Ullman, *Face recognition: The problem of compensating for changes in illumination direction*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1997. 19(7): p. 721-732.
- [54] Zou, X., J. Kittler, and K. Messer. *Illumination invariant face recognition: A survey*. in *IEEE International Conference on Biometrics: Theory, Applications, and Systems. pp 1-8*. 2007. Crystal City, Virginia, USA.

- [55] Otsu, N., *A Threshold Selection Method from Gray-Level Histograms*,. IEEE Transactions on Systems, Man, and Cybernetics, , 1979. 9(1): p. 62-66.
- [56] Lin, C. and K.C. Fan, *Triangle-based approach to the detection of human face*. Pattern Recognition, 2001. 34(6): p. 1271-1284.
- [57] Tou, J.T. and G. Rafael, *Pattern recognition principles*. 1974: Addison-Wesley.
- [58] Curran, K., X. Li, and N. Mc Caughley, *The use of neural networks in real-time face detection*. Journal of Computer Sciences, 2005. 1(1): p. 47-62.
- [59] Chang, Y.Z., K.T. Hung, and S.T. Lee, *Human face detection with neural networks and the DIRECT algorithm*. Artificial Life and Robotics, 2008. 12(1): p. 112-115.
- [60] Garcia, C. and M. Delakis, *Convolutional face finder: A neural architecture for fast and robust face detection*. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2004. 26(11): p. 1408-1423.
- [61] Naji, S., et al., *Neural Network-Based Face Detection with Partial Face Pattern*. The International Arab Conference on Information Technology, , 2011.
- [62] Gonzalez, R.C. and R.E. Woods, *Digital Image Processing 2ed*. 2002: Prentice Hall Press.
- [63] Kotropoulos, C. and I. Pitas, *Rule-based face detection in frontal views*. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997. 4: p. 2537-2540 vol. 4.
- [64] *CVL Face Database*, 2012.
- [65] *LFW Face Database*, 2012.
- [66] *JAFFE Database*
- [67] Schneiderman, H. and T. Kanade, *A Statistical Method for 3D Object Detection Applied to Faces and Cars*. Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2000. vol. 1: p. 746-751.
- [68] Hsieh, I.-S., K.-C. Fan, and L. C., *A statistic approach to the detection of human faces in color nature scene*. Pattern Recognition, 2002. 35(7): p. 1583-1596.
- [69] Srisuk, S., W. Kurutach, and K. Limpitikeat, *A Novel Approach For Robust, Fast And Accurate Face Detection*. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2001. 9(6): p. 769-779.