© International Journal of Combinatorial Optimization Problems and Informatics, Vol. 2, No. 1, Jan-April 2011, pp. 27-44. ISSN: 2007-1558.

Efficient Image Classification using Data Mining

Walid MOUDANI, Abdel Rahman SAYED

Lebanese University, Faculty of Business, Computer Department, Tripoli, Lebanon.
wmoudani@ul.edu.lb

Abstract. Skin region detection plays an important role in a variety of applications such as face detection, adult image filtering and gesture recognition. We present some definitions about digital imaging or numerical imaging and also the several image preprocessing and normalization. In this paper, we will tackle with two sub-problems which is the skin detection and face detection. We will propose skin detection method and some realized works in this area. Thus, we will detail the methods proposed for the face detection and the related studies found in literature. We will finish, by showing and analyzing the results obtained from tests made on a base of colored images with a complex background and make comparison to similar works which demonstrate that our algorithm achieve good accuracy.

Keywords. Detection of skin color, classification, face detection, face model, Data mining.

1. Introduction

Today, Information Technology (IT) is presented in virtually all areas. This leads to raise the importance of computers for remote monitoring which is the basis for control systems. Large volumes of databases, diversity and heterogeneity of data sources require a new philosophy of treating them. In this case, data mining looks to discover implicit knowledge in a dataset based on different techniques that can be implemented independently or coupled. These techniques aim to explore data, to describe their contents and extract the information more meaningful. Because much of the information that exists in organizations is informal and unstructured, these techniques are not limited to digital data and evidence, but must also address the textual and multimedia. In the recent years the acquisition, generation, storage and processing of data in computers and transmission over networks had had a tremendous growth. This changed radically with the appearance of the Internet and the first web browser which revolutionized the distribution of information. The ease of information exchange incited millions of people to create their own web pages and to enrich it with images, video, audio and music. Due to this rapid development in the domain of digital and informational data [20, 21], people now live in a multimedia world. More and more multimedia information is generated and available in digital form from varieties of sources around the world, this expansion presents new challenges.

Making shared data useful is a key interest for researchers and scientist, currently many text-based search engines are available on the World Wide Web and frequently used to search text documents, but searching for a multimedia content is not as easy because the multimedia data, as opposed to text, needs many stages of pre-processing to yield indices relevant for querying. Some techniques with different backgrounds have been proposed to solve such problem. As searching for multimedia data is a new domain and has been an active and fast advancing research area since the 1990s. During the past decade, remarkable progress has been made in both theoretical research and system development [22]. The first technique suggests assigning labels to multimedia data so data can be easily retrieved using text-based search engines, several approaches were proposed to use keyword annotations to index and retrieve images [24, 25, and 26]. Thus, the strategy of manually assigning a set of labels to a multimedia data, storing it and matching the stored label with the demanded label will not be effective. Besides, the large volume of video data makes any assignment of text labels a massively labor intensive effort, in addition, some media contents are difficult to be described in words and shapes cannot be expressed easily in textual form [23]. This approach is not satisfactory, because the text-based description tends to be incomplete, imprecise, and inconsistent in specifying visual information. Hence new techniques must be implemented and used for an efficient and effective result regarding multimedia data. The second technique suggests to automatically retrieving content of multimedia data, recent researches on image retrieval focus on content based image retrieval (CBIR) [40], which utilizes syntactic image features such as color, texture and shape to search image with similar contents based on examples they provide [34, 35, and 36]. Briefly, the retrieval process can be

Received Sep 10, 2010 / Accepted Nov 13, 2010

conceived of as the identification and matching of features in the user's requested pattern against features stored in the database and images which have a distance less than a specific threshold are retrieved [23]. This technique provides an efficient image search that can lead to retrieve exact result of the user request.

In this research, we introduce a new scope of data mining to image collections to define a model of skin effectively [7, 8]. For various reasons such as lighting conditions, ethnic diversity, etc. Identification of the skin is a complex problem. It seems interesting to apply the techniques of data mining in this context. As our study deal with two problems: skin detection and face detection, so this paper is organized as follows: firstly, a description of the problem is given, its difficulties encountered, its objective as well as the proposed solutions in the literature. Therefore, we provide a brief description of binary images and color spaces. However, a brief description of Data Mining as well as its power technique "decision tree" is presented. Thereafter, we present our methodology adopted in this problem starting by collection of data and preprocessing them in order to prepare them for the data learning phase. Firstly, we have proposed a model for skin and a model for face detection. We present the prediction model in order to deal with the considered problems here as well as its performance that is able to discriminate the pixels of skin from those of non-skin. Subsequently, we propose and present the techniques for face detection and segmentation of areas of skin region. Finally, we will discuss the design of our application and its environment. Tests and results are analyzed as well as conclusion and some thoughts on possible openings future work.

2. Problem statement

2.1 Introduction to the problem

Information more visual is a major consequence of this convergence between computing, Internet and broadcasting. More and more applications produce, distribute and use visual data, including still and moving images. The significant increase of visual information in the Internet and within organizations has been accompanied by an awareness of the importance of developing the tools to electronically process this information. This process can model, filter, classify, search and index this vast amount of data. If the digitizing of an image is a problem already solved technically, it is not well with the classification, cataloging and indexing images. Therefore, it is imperative to classify images according to their topics or their content. This classification will make a selection or access control according to the semantics and the type of images. This work is currently done manually by way of annotation, which has a cost in time and labor too great. Moreover, the size of image collections is increasingly enormous, making them virtually impossible manual annotation. Accordingly, we now find ourselves in front of a mass of visual information misclassified and uncontrolled Internet.

In our research, we are mainly interested in the problem of identifying skin pixels in the image. We have chosen a method based on detection of skin color approach. This research topic is significant issue insofar as it is necessary before considering testing and treatment in high level. Our goal is to build a generic model of skin which can serve as a basis for developing a detection system of human body parts (face, hand, etc..) to be of potential use for many applications such as applications related to security (surveillance, access control, etc..).

In addition to the identification of skin region, the detection and face recognition can also be used to facilitate search and navigation in a mass of videos. The establishment of such a system must, in advance, base on a detection and segmentation of skin regions. Another immediate application of our approach is the filtering of images on the Web. Indeed, the Internet appears as a huge reservoir of information including free access leads to undesirable practices such as children's access to adult sites. It is therefore necessary to introduce tools to filter sites for applications such as parental controls. These tools must be based on a semantic analysis of images in the process of identifying where the text alone is not sufficient. Thus, our research is to build a system able to discriminate the pixels of skin from those of non-skin, segment the image into regions of skin and detect regions of faces.

2.2 Difficulties

The detection of areas of skin or face is a very easy task for human: The image is captured by the eye which transfers it to the brain for analysis to verify the presence of skin and faces and locate them. This same task is very complex for a computer for many reasons [33]. We present below the specifications related to each one of the considered problems.

A- Detection of regions of skin:

- *Color of skin: Human have different skin colors (difference in the value of the pixel representing the skin of each person)*

- *Lighting conditions and illumination: In detection, the light is an important factor and is the most difficult problem to solve. It turned out that we can achieve a reliable system without taking this factor into consideration, hence the need for image preprocessing such as normalization and histogram equalization to reduce the effects of lighting and illumination.*

B- Face Detection:

- *Position: There are different positions of the face in an image (the computer will detect the face regardless of its position on the image).*
- *Size: The size of the faces are different from one image to another or from one person to another, hence the difficulty of implementing an algorithm that detects faces without consequences of this factor. There also the size of facial components such as nose, eyes, mouth, etc. which vary from one person to another, implying a greater number of parameters when performing detection (Figure 1).*
- *Presence or not of some components of the face: This is a mustache, beard, glasses, etc. which must be taken into consideration when implementing the algorithm.*
- *Occultation: A face that can appear in half in an image or sometimes partially obscured by an object requires us to define conditions for accepting the face by the system.*
- *Rotation: The faces are not always facing the camera. Some faces have a slope with a certain degree, which influences the detection system where the need to establish conditions to define a face candidate for acceptance.*
- *The complexity of the image detection can be on very complex images with many people in the same image, faces hidden or half hidden by objects with possibly complex backgrounds which increase the difficulty of detection.*

2.3 Literature overview

In this section, we focus on the presentation of the proposed solutions gathered in literature in order to tackle with the two sub problems considered here. Firstly, concerning the sub problem of the detection of skin areas, several types of methods for discriminating between skin pixels and non-skin are found in literature. These methods can be divided into three categories: *parametric methods*, *non-parametric methods* and defining explicitly the *skin as a cluster* [10, 11, 12]. The parametric Gaussian model is based on the use of a Gaussian model defining the distribution of skin color in a color space given [1]. The non-parametric method estimate the skin color histogram of the training set without the prior creation of a skin model [2, 27, 31, 32]. The method "Explicit Skin Cluster" consists essentially in defining the skin pixels by threshold giving details of the range of flesh color in a appropriate color space. This method is very popular related to its simple implementation and also because it requires no prior learning. Secondly, concerning the detection of faces sub problem [19], several types of methods are found in literature whose main features are:

- Method based on knowledge (Knowledge based method): This method is based on rules that represent the core components and represent human faces. The rules are usually made from the relationship between facial features. This method is mainly used for face detection. For example, the faces on the images often have two eyes are symmetrical, nose and mouth [28, 29]. The relationship between these members may be represented by the distance between them and their position. One problem with this approach, namely the difficulty of translating knowledge of the human face to clear rules, which may cause detection errors and make the system unreliable.
- Feature invariant (invariable characteristics): This algorithm aims to find structural features even if the face is in different positions, light conditions or change of angle [15]. The problem with this method is that image quality can be severely diminished because of illumination, noise or occlusion. However, there are several properties or invariant features of the face, the main ones include:
    - *Facial feature (characteristic of the face):* This method uses the plans edges (Canny detector) and heuristics to remove all groups except those edges which represent the contours of the face. An ellipse is deduced as the boundary between the background and face. It is described as consisting of point of discontinuity in the function of luminance (intensity) image. The basic principle is to recognize objects in an image from contour models known in advance. To accomplish this task, two methods are presented: the Hough Transform and Hausdorff distance. The Hough Transform is a method to extract and locate groups of points within certain characteristics [16]. For example, the specific searches can be straight lines, arcs, shapes, whatever, etc. In the context of face detection, it is represented by an ellipse on the card edges. The application of circular Hough transform then produces a list of all candidates being circles or derivatives. The basic algorithm has been modified to appear and show several variants, including the Randomized Hough Transform, which can be applied to research of arbitrary forms like circles or ellipses (exp: faces). Finally, the Hough transform can be used to detect the eyes and irises. Contrary, this method will fail when the image is too small or when the eyes are clearly visible. Concerning the method of Hausdorff distance [11], it uses for its edges as data and a special algorithm for Template Matching [11]. Indeed,

the Hausdorff distance is designed to measure the distance between two sets of points, which are mostly card edges (image search) and a model. The basic algorithm performs the search of the best places to match anywhere in the image (translation) and for different rotations. This research may also include a scale factor to detect variations of the original model. This method has been used successfully to detect faces with frontal views. Adapting it to overcome various poses (axial rotation of the head), however, brings problems, because different models should be used. In addition, a complex stage of decision would be mandated to separate the false detections and multiple detections.

- *Texture:* The texture of the human being is separate and can be used to separate faces from other objects [17]. In papers ([17, 18]), a method of detecting faces in an image based on the texture have been developed. The texture is calculated using the characteristics of second-order sub 16 × 16 pixels. Three types of characteristics are taken into consideration: skin, hair, and the remaining components of the face [30].
- *Skin color (color):* The skin color of humans was used for the detection of faces and its relevance has been proven as a feature specific to the face. However, the color is different person to different people and their origin. Several studies have shown that the greatest difference lies largely in intensity than chrominance [3, 4, 5, 8, 37, 38].

3. Background on the digital images and color spaces

In this section, a brief presentation of digital images is presented. Thus, the color spaces evoking the representation of colors in the image, the primary colors systems and the different color spaces are presented.

3.1 Digital Images

3.1.1 The concept of pixel

An image consists of a set of points called pixels (pixel is short for Picture Element). A pixel is the smallest component of a digital image. All of these pixels are contained in a two-dimensional array forming the image (Figure 2). Given that the screen performs a scan from left to right and from top to bottom, generally known by the coordinates [0, 0] the pixel in the top left of the image, this means that the axes of the images are oriented as follows: the X axis is oriented from left to right, and the Y axis is oriented downwards, contrary to conventional mathematical notation, where Y axis is directed upwards.

3.1.2 Bitmap and vector images

There are generally two broad categories of images:
- Bitmap images: it is a set of pixels in an array, each of these points having one or more values describing its color (Figure 3.a).
Vector images: images are vector representations of geometric entities such as a circle, rectangle or segment. These are represented by mathematical formulas (a rectangle is defined by two points, a circle with a center and a radius, a curve by several points and an equation). It's the processor who will translate these forms in information interpreted by the graphics card (Figure 3.b).

Because a vector image consists only of mathematical entities, it is possible to easily apply geometric transformations (zoom, stretch), while a bitmap image made of pixels, can suffer such changes only by a loss of information, called distortion. We called by aliasing the occurrence of pixels in an image following a geometric transformation (including expansion). In addition, vector graphics (called clipart when it is an object vector) used to define an image with very little information, which makes the files very light. Instead, a vector image can only represent simple forms. While a superposition of various simple elements can yield very impressive results, any picture cannot be rendered vectorally, as is the case of realistic picture.

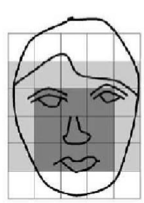


Figure 1. Face details

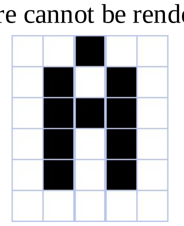


Figure 2. a two-dimensional array forming the image

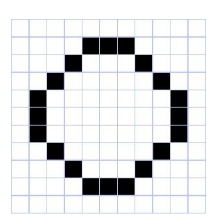


Figure 3.a. bitmap image

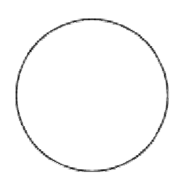


Figure 3.b. Vectorial image

The image "vector" above is a representation of what might look like a vector image, because image quality depends on the material used to make it visible to the eye. Your screen can probably see this image at a resolution of at least 72 pixels per inch, the same file printed on a printer would give a better picture quality because it would be printed at least 300 pixels per inch.

3.1.3 File Format

We have seen previously how an image is encoded for display on a monitor, however, when we want to store an image in a file format that is most convenient. We may indeed want an image that takes up less space in memory, or an image you can enlarge without showing any pixilation. Thus, it is possible to store the image in a file with a data structure describing the image using the equation, which must be decoded by the processor before the information is sent to the graphics card. Concerning the types of file formats, there are many file formats. Among the graphic file formats, the list below contains the commonly used (Table 1):

Table 1. List of the types of file formats

Format	Compression	Maximum dimensions	Maximum number of colors
BMP	none / RLE	65 536 x 65 536	16 777 216
GIF	LZW	65 536 x 65 536	256
IFF	none / RLE	65 536 x 65 536	above 16 777 216
JPEG	JPEG	65 536 x 65 536	above 16 777 216
PCX	none / RLE	65 536 x 65 536	16 777 216
PNG	RLE	65 536 x 65 536	above 16 777 216
TGA	none / RLE	65 536 x 65 536	above 16 777 216
TIFF / TIF	Packbits / CCITT G3 & 4 / RLE / JPEG / LZW / ITU-T	$2^{32}-1$	above 16 777 216

3.2 Different primary color systems and color spaces

The color information is one of the most important information in the field of digital image processing. We can consider the color of a pixel as a vector in three dimensions. We note that the search for optimal color spaces is well developed. As the choice of color space to use, significantly influence the results, hence the need to choose carefully. The information is much more significant for the detection of regions. Thus, we will discuss in the following different systems of representation of the color [39].

The primary systems are based on a theory that any color can be identically reproduced by the algebraic combination of three colors called primaries. This implies that we can reproduce a color from a mixture of both. There are several primary systems which are commonly used; the difference among them is that each system is defined by reference to a very specific illuminant. We note for example: CIE International Commission on Illumination (illuminant E), NTSC: National Television Standards Committee (illuminant C), EBU: European Broadcasting Union (illuminant D 65). The transition from one system to another is always ensured by the existence of a transition matrix P and a transition of inverse matrix inverse P^{-1}.

Concerning the space color, there are several model such as: The RGB model (this model is the basic principle of television monitors and screens scanning, since it is by superposition of red, green and blue as the color display is achieved), the YCrCb model (used in the JPEG standard), the YIQ model (This is a recording of RGB system by NTSC (National Television Standards Committee)), the HSV model, and the CMY model.

4. Data Mining

4.1 Introduction to the Data-Mining

The 1990s has brought a growing data glut problem to many fields such as science, business and government. Our capabilities for collecting and storing data of all kinds have far outpaced our abilities to analyze, summarize, and extract “knowledge” from this data ([41], [42], [43], [45]). Traditional data analysis methods are no longer efficient to handle voluminous data sets. Thus, the way to extract the knowledge in a comprehensible form for the huge amount of data is the primary concern. Data mining refers to extracting or “mining” knowledge from databases that can contain large amount of data describing decisions, performance and operations. However, analyzing the database of historical data containing critical information concerning past business performance, helps to identify relationships which have a bearing on a specific issue and then extrapolate from these relationships to predict future performance or behavior and discover hidden data patterns. Often the sheer volume of data can make the extraction of this business information impossible by manual methods. Data mining is a set of techniques which allows extracting useful business knowledge, based on a set of some commonly used techniques such as: Statistical Methods, Neural Networks, Decision Trees, Bayesian Belief, Genetic Algorithms, Rough Sets, and Linear Regression, etc. ([44]]). We focus our interest on decision tree technique since it is used in our solution approach. A decision tree is a tool for decision support that divides a population of individuals into homogeneous groups based on attributes discriminating on the basis of an objective set and known. It can make predictions based on known data on the problem by reducing, level by level, field of solutions [9]. Each internal node of a decision tree involves a discriminating attribute on elements to be classified that divides these elements seamlessly between different son of this node. This method has the advantage of being readable and allows analysts to identify couples <attribut, value> discriminants from a large number of attributes and values.

4.2 Principle of learning a decision tree

With input, a sample of m classified records $(\vec{x}, c(\vec{x}))$, a learning algorithm must provide as output a decision tree. Most algorithms proceed from the top down, that is to say they choose the root of the tree (usually a test) and then, recursively, choosing the label of son. To simplify the presentation, we limit ourselves in this section to problems where the attributes are discrete and the number of classes equal to 2. The generic algorithm can be written (Figure 4):

```
Given: a sample S of m records ordered (x, c(x))
Initialization: empty tree; current node: root; cu
sample: S
Repeat
if the current node is terminal then
label the current node by a leaf
otherwise
select a test and create the subtree
endif
current node: a node not yet examined
current sample: sample reaching the current node
until production of a decision tree
prune the decision tree obtained
Output: decision tree pruning
```

Figure 4. Learning algorithm for decision trees

It is possible to continue the growth of the tree until getting a tree with zero error (if possible: if there are no examples of the same description but different classes) or a tree with error measured on the smallest possible training set. However, the objective of a classification procedure is to properly classify examples not yet met; we talk about power of generalization. If the algorithm provides as output a very large tree that classifies well training sample, we face the problem of over-specialization. Thus, generating a complete tree is very difficult and tackle the performance of the algorithm; then we have applied the pruning of the decision tree in order to obtain a smaller tree (branches are pruned, that is to say that we destroyed sub-trees) in order to obtain a tree with better able to generalize (even if it increases the error on the training set).

4.3 Rating the classifiers

Our goal is to find the best classifier that is best suited to our problem. It is considered that no one learning method that outperforms the others, because each method has its strengths and weaknesses. It is recommended to test several methods to find the best of them or make them cooperate. The selection of models is one of the most lucrative fields of experimental analysis which allows defining objectively the best method on the given data. It is difficult to formulate general indicators to validate the models, in most cases, researchers are working on the error rate because it is the only indicator that is truly comparable to an algorithm to another, but of course there are also other criteria such as complexity and response time. In general, the performance of a model is assessed through a confusion matrix that compares the actual situation and the situation predicted by the model to estimate the error rate.

A- <u>Confusion matrix:</u>

The confusion matrix appears as a contingency table comparing the assignment class (column) with the original class (line) of individuals in the sample (Table 2):

Table 2. Confusion matrix for a model of 2 classes A and B.

	Assignment Class	
Initial Class	**A**	**B**
A	$N_{A,A}$	$N_{A,B}$
B	$N_{B,A}$	$N_{B,B}$

In this matrix, $N_{A,B}$ is the number of cases of class A assigned in Class B and N_{BA} represents the number of cases of the class B affected to class A, while N_{AA} and N_{BB} represent the correct number of classification. From this confusion matrix we can identify three types of indicators: Accuracy, Recall, and Precision.

B- Cross-Validation:

Cross-validation suggests dividing equally the sampling dataset in “s” parts where the (s-1) parts are considered as the learning parts of the database and the remaining part for the test. Then, there is a permutation of bases tested, and thus a table of confusion is created by averaging the "s" testing. It is therefore a repetition of the couple "learning-validation", but ensuring that there is no overlap between the validation samples.

C- The Bootstrap

The idea is to use the sample of observations to allow statistical inference finer. It makes a number of samples, termed bootstrap sample obtained by random observations of the initial sample. In each bootstrap sample, we estimate the parameters of the model. Therefore, there is a sequence of parameters. Under certain regularity conditions, the theory shows that the distribution of the following parameters obtained converges to the true distribution of the parameter. The bootstrap was imposed today in statistics as a convenient technique of statistical inference. It requires, indeed, "few" assumptions and is relatively easy to program, they are, in fact, randomly draws. However, the big drawback is the significant computational capabilities that the application of these techniques requires; more often it is recommended that at least one hundred repetitions to expect a good reliability.

5. Segmentation of the image into regions of skin

In this section, we detail the various construction phases of our skin model, relying on techniques of data mining and image analysis. The method uses data mining techniques to produce the prediction rules based on the RGB color space, followed by a phase of identification and segmentation into coherent skin regions using the rules already generated.

5.1 Presentation of the proposed approach

The preparation of the skin model was conducted through three stages (Figure 5):
1- The first step is devoted to the composition of our data set, and the preparation of data for the learning phase.
2- The second step is to find a prediction model associated with a representation space capable of discriminating the pixels of skin from those of non-skin. The representation space can be a conventional color spaces, color space is chosen as the RGB color space. The data mining tools allow us then to retain adequate decision rules.
3- The last step is the segmentation of the image and identification of different areas of interest (skin) taking into account the rules extracted during the previous phase.

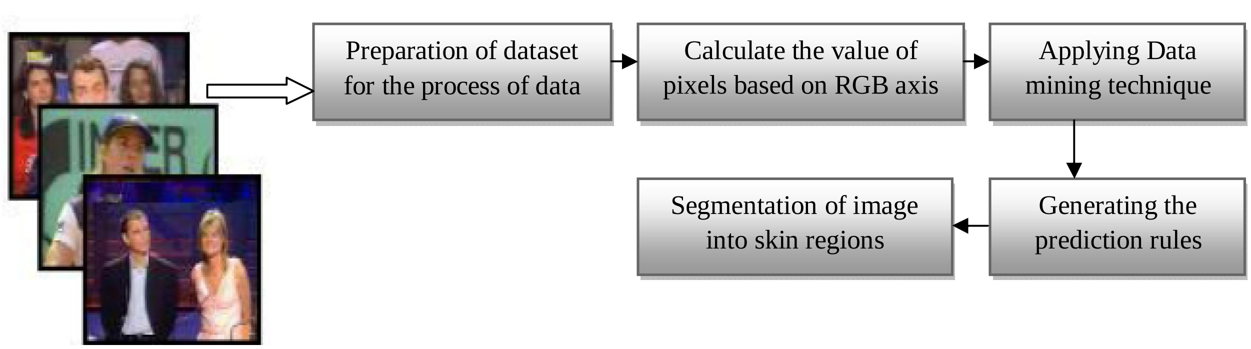


Figure 5. Flowchart of the proposed approach

5.2 Construction of Dataset

The construction of the learning base is an important element in a process of extracting knowledge from data. For our classification problem of pixels skin or not, the quality of the dataset can be judged on the following factors: The size of the database and variety in the content of images, which should be representative for the different sexes, races, and lighting conditions. We have used a dataset composed of 100 images for different sexes, races, and lighting conditions, unlike most existing work which have a learning curve on predefined classes of images under any lighting condition known in advance. This dataset led to a model consisting of ≈ 60,000 pixels. The construction of the training dataset is not a simple task. And as the size of the database must be large, it is difficult to construct it manually. Hence the need for such software to help us in building this database. For this reason, we have developed a tool that allows us to perform this task. The construction of dataset using the developed tool has two phases:

Phase I: Recording pixels in database SQL Server
This is done simply by clicking with the right mouse button to start recording pixels. At the beginning, we choose the skin option in the tool, thereafter; the pixels are stored as pixels of skin during movement of the mouse on the image. Otherwise, the pixels are stored as non-skin. To complete the recording pixels, we just click with the right mouse button.

Phase II: Creating Weka data file (. ARFF)
In the second phase, we focus on the preparation of the data file that will be used in the available tool WEKA ("Waikato Environment for Knowledge Analysis"). So, the data file should have a format understandable by WEKA. This tool is an open source developed using Java. It contains a different set of techniques such as: classification, segmentation, association rules, etc. Also, a set of methods for each technique is listed to be used by the user upon the considered case. Thus, the aim of this tool is to manipulate and analyze data files, implementing the most artificial intelligence algorithms (decision trees, neural networks, etc.).

5.3 Skin model

5.3.1 Identification of skin pixels

A- Learning File
We define for each class skin or not skin, a set of representative N pixels P_i *(i = 1… N)*. These representative pixels are extracted from the images of learning which contains pixels of the skin and not skin. In the RGB color space, we characterize each pixel P_i, representative of each class (skin or no skin), with an observation $X_i^j = \left[X_i^r, X_i^G, X_i^b, X_i^c\right]$ where X_i^r is the red component, X_i^G is the green component, X_i^b is the blue component, X_i^c is the class of the pixel of index i. Knowing the methods of data processing often require a particular format of data files, we state all the observations. The aim is to construct decision rules that can be written as follows:

If (Condition $_{1,1}$ **and** Condition $_{1,2}$ **and** Condition $_{1,3)}$ => **Skin = FALSE**
… … … … … …
If (Condition $_{k,k1}$ **and** Condition $_{2}$ **and** Condition $_{k,3)}$ => **Skin = FALSE**
… … … … … …
If (Condition $_{N,N1}$ **and** Condition $_{2}$ **and** Condition $_{N,3)}$ => **Skin = FALSE**
If no => **Skin = TRUE**

Where Condition $_{i,1}$ is written as (R> … or R<...); Condition $_{i,2}$ is written as (V> … or V<...); Condition $_{i,3}$ is written as (B> … or B<...). (i = 1 .. N; N is the number of conditions where a pixel is interpreted as a not skin pixel).

B- Extracting decision rules
To find the best predictive model, we have tested several techniques based on induction graphs J.48 and rules. JRip using WEKA 3.6.0 and we get the following results (Table 3):

Table 3. Comparative results issued from two decision tree algorithms

Algorithm	Cross-Validation	Confusion Matrix
J48 decision tree	*Correctly Classified Instances 5573 96.1857%* *Incorrectly Classified Instances 221 3.8143%* *Kappa statistic 0.8698* *Mean absolute error 0.0535* *Root mean squared error 0.1849* *Relative absolute error 18.1438%* *Root relative squared error 48.1385%* *Total Number of Instances 5794*	*a b <- classified as* *4651 101 | a = TRUE* *120 922 | b = FALSE*
Association rules **JRip**	*Correctly Classified Instances 5550 95.7887%* *Incorrectly Classified Instances 244 4.2113%* *Kappa statistic 0.8563* *Mean absolute error 0.0642* *Root mean squared error 0.1956* *Relative absolute error 21.7589%* *Root relative squared error 50.9324%* *Total Number of Instances 5794*	*a b <- classified as* *4639 113 | a = TRUE* *131 911 | b = FALSE*

5.3.2 Segmentation of the image into regions of skin

Many applications using a model of skin need knowledge of areas of skin. Also, we conduct a segmentation stage whose objective segmentation procedure is to extract meaningful regions representing objects of compounds of skin pixels. For this it needs a good segmentation method that is able to identify consistent areas both visually and in terms of their content to facilitate their interpretation. In the following sections, we present the technique of image segmentation based on the labeling of connected components. We present here the proposed algorithm (called SkinRegionCounter Algorithm) aiming to segment the image, as:

Phase I: Preparation of the image
The approach used in this phase is to replace the skin pixel by white pixel and those of non skin by black pixels. The result is an image containing only two colors, black and white. The color black is used as separator areas of skin (white) in the next phase. The algorithm corresponding to this phase is as follows (Figure 6, 7):

```
Create the List "[K]" of pixels of an image.
For each pixel P in List "[k]"
        IF Skin(P) Then
                Change the color of P to White
      Else
                Change the color of P to Black
        End IF
End For
```

Figure 6. Algorithm corresponding to the preparation of image

Figure 7. Preparation of image for the segmentation phase

Phase II: Marking connected components
The approach used in this phase is the final award on the image of the same label to all points of the same object (white region). The algorithm used is as follows (Figure 8, 9):

```
//Form(P): is a point object (i.e. I(P) = (255)).
//Neighbors(P): P has at least a point object in its neighborhood.
//LabelNeighbors(P): gives the label of its neighborhood.
 //Pred(P): is a list of predecessors of P.
//MinPred(P): is the minimal label of its predecessors.
//AssignLabel(e, P): assign the label e to P and Modify the  list of correspondences.
//NLab(): give a new label.
----------------------------------------------------------------------------------------------------------------
Create List "[K]" of image pixels.
Créer List "[L]" of correspondances

// Having the number of pixels as length
//The elements of this list are couples as (Pixel, Label).

For each pixel  P  in list "[k]"
            IF  Form (P) Then
               IF Neighbors (P) Then
                     AssignLabel (LabelNeighbors(P), P)
               else
                        AssignLabel(NLab(), P)
                End IF
          End IF
End For
```

Figure 9. Labeling of connected components

Figure 8. Algorithm corresponding to the labeling of connected components

<u>Phase III:</u> Building the list of connected components (regions of skin)
The algorithm used in the current phase is the following (Figure 10):

```
//SameAs(P, [L]): Give the set of elements of list L having the same label as P.
//DelEls(P, [L]): Delete all elements of the list L having the same label as P.
//CreateComp([L]): create a connected object (skin region) and ad dit to list "[K]".
// FirstElement([L]): Giev the first element of the list.

Create the list "[L]" of matches.
// /Having as length, the number of pixels, using the previous algorithm.
//The elements of this list are represented as couple (Pixel, Label).

Create the list "[K]" of connected matches

// Having as length the number of labels used.

While "[L]" is not empty do
P=FirstElement([L])
CreateComp(SameAs(P, [L]))
DelEls(P, [L])
 End
```

Figure 10. Algorithm corresponding to the labeling of connected components

6. Face detection problem

6.1 Introduction

Face detection is used to recognize an object in an image as a face and distinguish it from the rest of the image. The detection of human face is a complicated task for the face may be a pose, size, color and facial expression variables. In addition, other factors, such as wearing glasses, the presence or absence of hair, and occlusions can be unpredictable appearance of the face. Other effects, such as lighting conditions and background variables can make the generalization algorithms for face detection difficult [6].

The detection of human face in a given image is an important task in many applications of computer vision, such as identification of the face, the human-machine interface, etc. Research in these fields focus on improving the information exchanged between human and machine. This requires the integration of new media for a natural dialogue between man and machine provided with tools to capture not only inconvenient for the user to perform spontaneous movements. Thus, the use of visual information about the user including gestures and facial expressions (happy, sad, etc...) help to improve communication in several senses. The system detects the presence of a user. It then locates his face, his nose and both eyes, and he follows in the sequence of images. Therefore, the success of these treatments depends heavily on the detection phase.

6.2 Elaborating the face model

Our model was created while specifying the most important characteristics of face region. From these characteristics, we can decide whether a skin region represents a face or not.

6.2.1 Characteristics of regions face

An area of skin containing at least two non-skin regions. (Eg eyes) (Fig 11.a). The ratio of area surface must be between certain values (Figure 11.b). There is a distance d belongs to interval [d0, d1] between two sets of non-skin regions in this region. (The eyes should be distributed between the left and right side of the face for example) (Figure 11.c). In addition, non-skin regions should exist in the upper face. The distance between the ordinates of the centers of gravity of non-skin regions of the current region must belong to a certain interval (Figure 11.d).

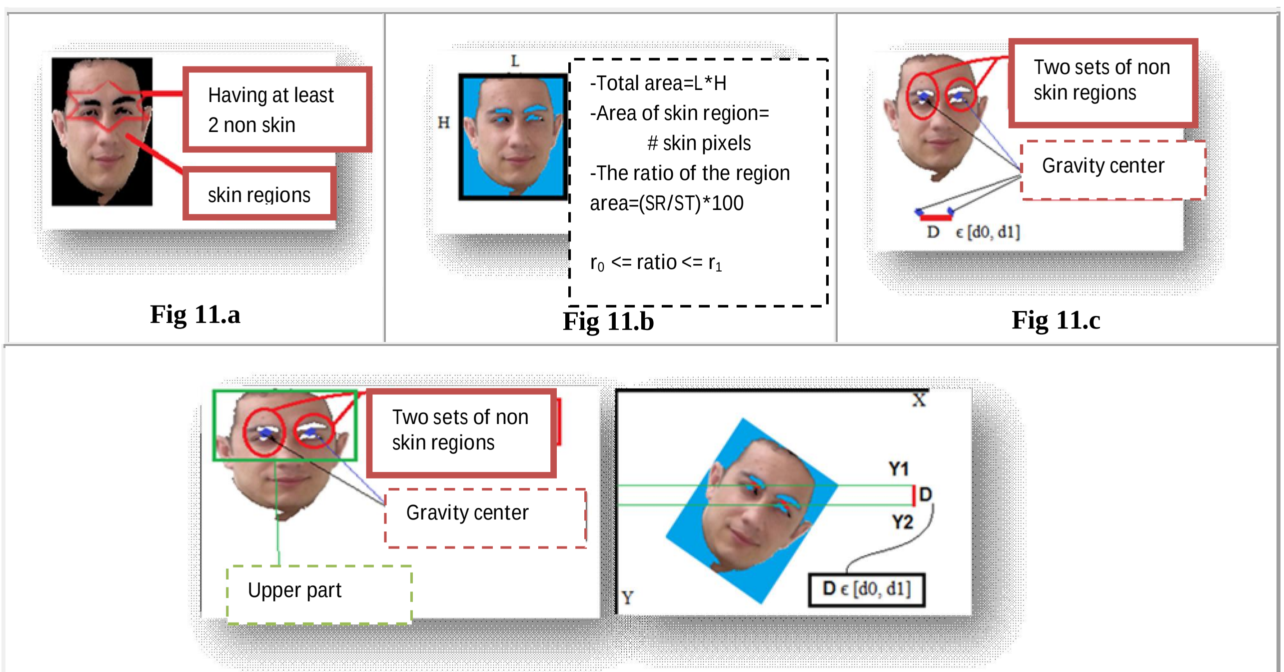


Fig 11.a

Fig 11.b

Fig 11.c

6.2.2 Method of Face Detection

The proposed approach is based on the use of the detection method for removing sections that did not match skin color; and on the algorithm of face detection to remove sections lacking the characteristics of a face. The main steps are summarized as follows: *Replacement of non-skin pixels in the black color from an input image, Segmentation of image in areas of skin, and Applying the algorithm of face detection.* Our proposed approach for face detection has 2 phases:

Phase 1: segmentation of the image into regions of skin.

The approach used in this phase is to segmenting the image into regions of skin. The algorithm used in this phase is SkinRegionCounter.

Figure 12. Segmentation of image into skin region

Phase I1: Detection of face

The algorithm used in this phase scan all areas of skin one by one, checking if the current region is a face or not. The algorithm is written as follows (Figure 13):

```
Create the List "[K]" of skin regions.
Create the List "[V]" for faces regions.

FOR each region  R  in List "[k]"

        IF    ISFACE(R) THEN
                Add R to List "[V]"
        END IF
END FOR
```

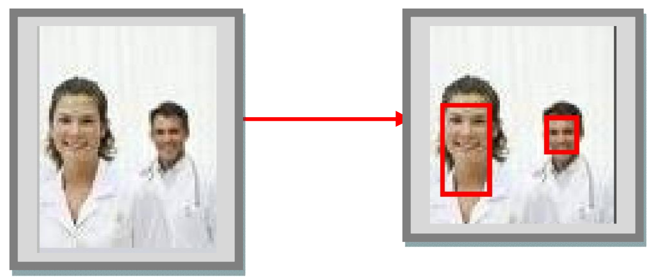


Figure 13. Algorithm for face detection

7. Design and Implementation

After presenting the different techniques of image segmentation into regions of skin and the technique of face detection, we discuss in this section the design of our application. Also, we will describe the environment on which we made our application.

7.1 The environment of our application

To achieve our project and estimate the effectiveness of the chosen methods “Skin Color”, segmentation of the image into regions of skin and face detection, we used Microsoft C #. NET to program the modules of our application. To better present and better describe the various steps proposed in the design of our system, we'll see the system from two different views: a user system view and designer system view.

A- User Viewpoint

The system of image segmentation into regions of skin, image classification and face detection is handled as follows: Each user must load an image for processing. The load may be from several sources, file, memory or a screenshot. We also offer the option to use or not a pretreatment to improve the image quality and the quality of detection of skin color. The user can also set more technical parameters related to image classification and filtering of regions of skin. Finally, the user can introduce an image containing several faces with complex backgrounds to achieve segmentation and detection of various existing faces (Figure 14.a).

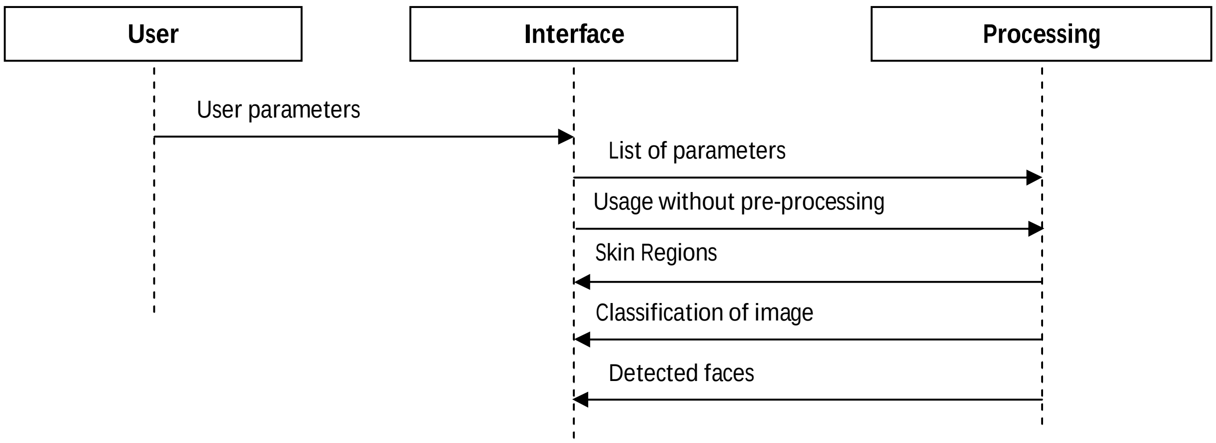


Figure 14.a. Interaction between the user and the system.

B- Designer Viewpoint

B.1 Platform Architecture

We have introduced in previous sections the different techniques and methods for segmentation, classification and face detection. This introduction has helped us to implement our architecture with the appropriate methods. Our study is a research area not yet fully explored, especially in the case of image processing in many faces with the constraint of random and complex backgrounds. We observe that in this sector, there are two research centers, improving the detection rate or improving the execution time. We have dedicated product architecture organized in series to improve the detection rate to be presented as follows (Figure 14.b):

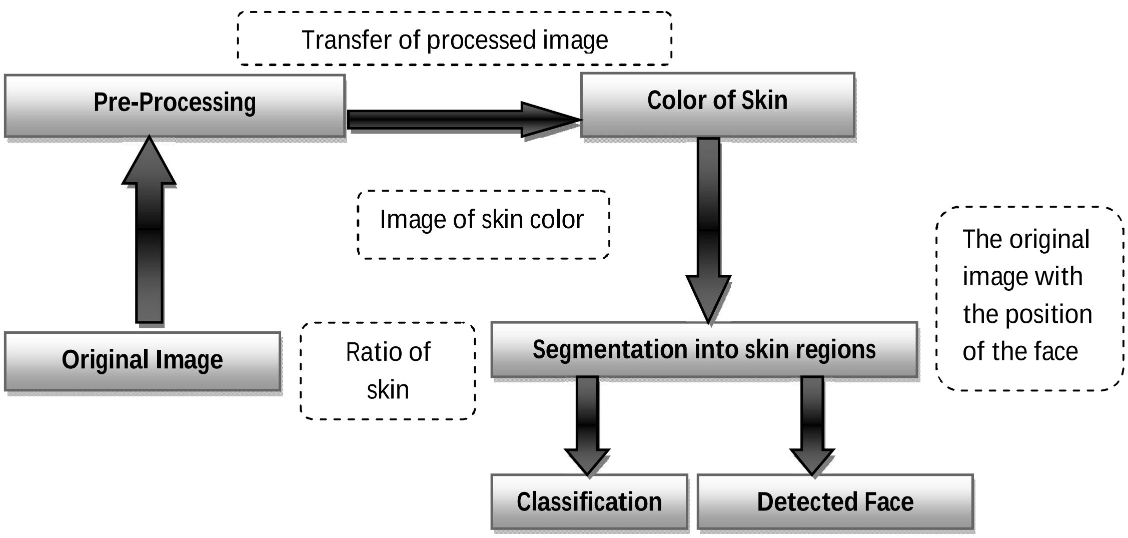


Figure 14.b. The detailed steps of our system.

B.2 The pre-processing (Normalization of an image)

Each image can be built from a various level of light or with different lighting and shadows. It is therefore important that the data are standardized and that the contrast variations are representative of facial features and not its environment. The normalization of images is done in two stages: Equalization of light in the picture and Histogram equalization. But in our preprocessing, we need to equalize the light in the image of "equalization of contrast" because this step significantly improves the resulting detection.

B.3 Equalization of background light intensity

The first stage of this filter of normalization is the standard equalization of intensities encountered in the image window. We will consider the pixels not hidden by the mask, the background is irrelevant. The principle is to define a function that approximates the overall intensity of the window to its mean. This will effectively keep the local variations of intensity but reduce the overall average intensity by a constant region. Thus, the influence of initial exposure of the image is much attenuated. This allows, on the first criterion, standardization of images from heterogeneous environments and enables the comparison (Figure 15.a, 15.b).

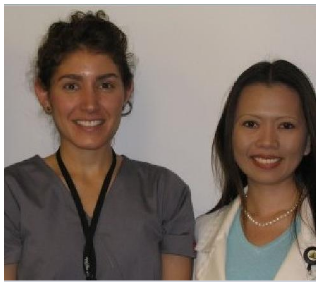


Figure 15.a. Original image

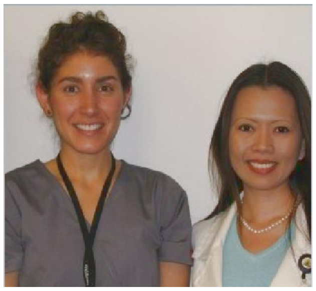


Figure 15.b. Processed image

B.4 Skin color based on decision rules extracted.

Then, we have the original image at the entrance of the module that is represented in RGB color space; we traverse the image in pixels while checking the following conditions, as example:

```
if (R <= 100) return false;
if ((B> = 178) & & (R <= 225)) return false;
if ((R <= 136) & & (G> = 139)) return false;
if ((R <= 82) & & (G> = 67)) return false;
if ((B> = 229) & & (R <= 249)) return false;
if ((R <= 124) & & (B <= 106) & & (G> = 101)) return false;
if ((B <= 53) & & (R> = 117)) return false;
if ((R <= 156) & & (G> = 129) & & (B <= 141)) return false;
if ((B> = 215) & & (G> = 255)) return false;
if ((B <= 63) & & (G> = 105) & & (R <= 155)) return false;
if ((G> = 146) & & (R <= 174) & & (B <= 130)) return false;
return true;
```

7.2 Tests and results

A series of experiments will be presented in this section. Tests were performed on color images containing faces with any background and complex conditions.

7.2.1 Data preparation and parameters settings

The images dataset of test used consists of 100 images of one or many faces with complexes backgrounds and lighting conditions. Our application is designed to segment the image into regions of skin, classify the images and detect faces in extreme conditions.

In order to achieve our goal, several parameters must be set (Figures 16.a, 16.b). In figure 16.a, the filter chosen from this menu will be applied to the image before the start of phase of segmentation and face detection. Before the start of segmentation and face detection phase, the user can determine areas of skin that he wants to display by specifying the dimensions (Figure 16.b). By default, all regions are displayed. The value 0 of a given attribute means that there is no condition on it. MinWidth (resp. MaxWidth): the region with a width less (resp. great) than this value will be ignored during segmentation of the image. MinHeight (resp. MaxHeight) the region with a height less (resp. great) than this value will be ignored during segmentation of the image.

Figure 16.a. Settings the parameters of pre-processing

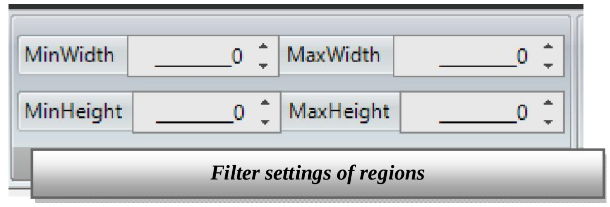


Figure 16.b. Settings the parameters of regions segmentation

To make our system flexible, the user can define classes of images and their thresholds. It can depends on the ratio of skin in the image. Also, we can set the parameters of face detection to define the face model based on the following attributes: interval surface area of face, interval number of non-skin regions in the region face, rotation interval of face, differences between the ordinates of both eyes, and minimum distance between eyes.

7.2.2 Different tests

We have conducted tests with the same parameters mentioned above on a sample of 100 images (Figure 17) and here are some results (Table 4):

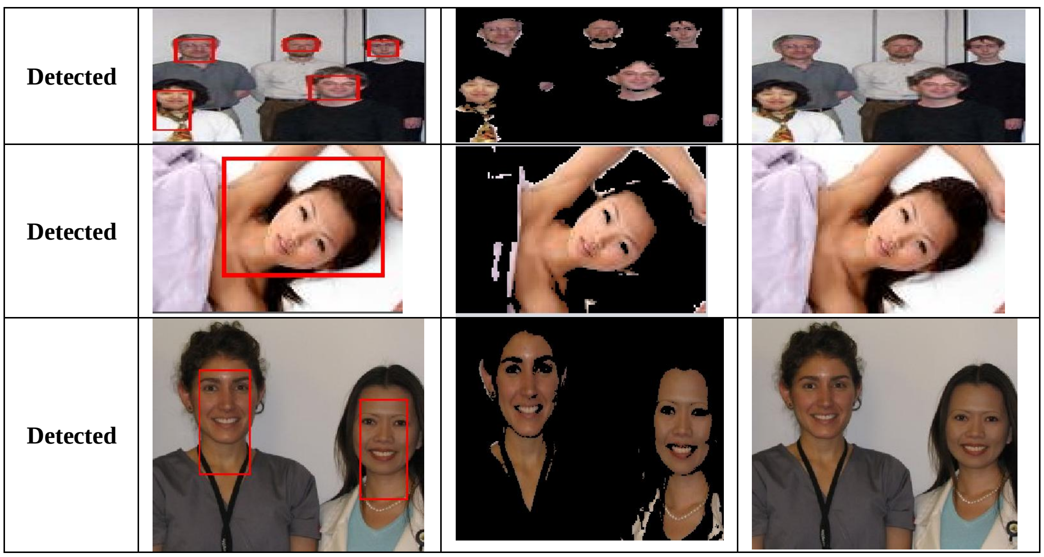


Figure 17. Extract from the database

Table 4. Results of experiments

Not Detected	False detection	Faces Detection	False classification	Classification	Number of images
4	2	94	7	93	100

7.2.3 Comparison with other software developed in the same field

We made comparisons between our software and another met on the Internet (Real Time Face Detection Demo V2.0) (Figure 18).

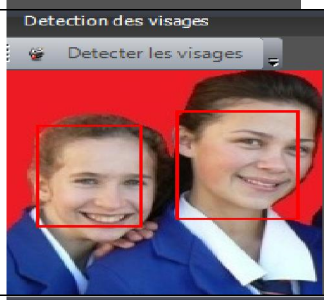
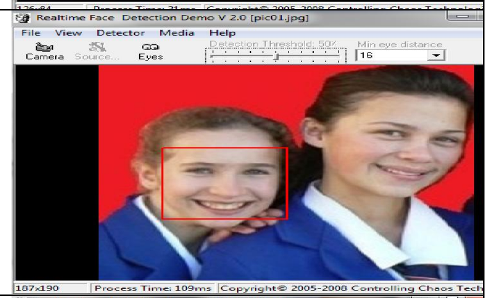
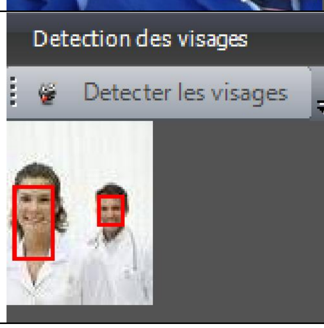
Proposed system		**Tool existing in literature (Real Time Face Detection Demo)**	
Supplementary Detection *Requesting time : [85 ms]*		*Not Detected* *Requesting time :[57 ms]*	
Detected *Requesting time : [112 ms]*		*Not Detected* *Requesting time : [97 ms]*	
Detected *Requesting time : [150 ms]*		*Partial Detection* *Requesting time : [109 ms]*	
Detected *Requesting time : [114 ms]*		*Not Detected* *Requesting time : [31 ms]*	

Figure 18. Results of detection faces with different conditions

9. Conclusions and perspectives

Detection of skin color in color images is a very difficult especially because it is considered a preliminary step in most systems of human computer interaction, motion detection, or faces, filtering adult sites. Moreover, the performance of these systems is closely linked with the results obtained in this step. In this research, we have proposed an approach for extracting knowledge for the development of a skin model with application including segmentation, image classification and face detection. To illustrate the robustness of our approach, the model of skin derived from our work has been successfully used in two applications. The first application concerns the detection of faces in color images. The second application concerns the classification of an image to be human, adult sites or other. Regarding the application of face detection, one of the main drawbacks of the method is the execution time which is very important because the browse of pixels in the image multiple times for each region is very costly in time of calculation. We obtained in this work, encouraging results with different configurations; we also give the possibility to vary the parameters as needed to users.

We also offer such opportunities, the integration of visual content analysis, based on skin color, for classification and filtering of Web sites that are becoming increasingly visual and multimedia.

Finally, as we stated previously, the face detection is the first phase of the system of face recognition. A perspective that results is the integration of our platform in a biometric product. Also one of important points is to encourage developers to work on improving performance in terms of computing time of the used methods.

References

[1] Yang, M.H., and Ahuja, N.: Gaussian Mixture Model for Humai Skin Colour and its Applications in Image and Video Databases, Storage and Retrieval for Image and Video Databases (San Jose, January 1999) 458-466.

[2] Joenes, M. and Rehg, J.: Statistical Color Models with Application to Skin Detection IEEE Conference Computer Vision and Pattern Recognition, CVPR'99 (1999) 274-280.

[3] Cheng, H.D., Jiang, X.H., Sun, Y. and Wang, J.: Color image segmentation: advances and prospects, Pattern Recognition, Vol. 34, No. 12(2001) 2259-2281.

[4] Orchard, M.T. and Bouman, C.A.: Color quantization of images, IEEE Trans. on Signal Processing, Vol. 39, No. 12 (1991) 2677-2690.

[5] Balasubramanian, R., Allebach, J.P. and Bouman, C.A.: Color-image quantization with use of a fast binary splitting technique, Journal of Optical Society of America, Vol. 11, No. 11 (1994) 2777-2786.

[6] Garcia C. and Tziritas G.: Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis, IEEE Transactions on Multimedia, Vol. 1, No. 3 (1999) 264-277.

[7] Lemieux, A.: Système d'identification de personnes par Vision numérique (December 2003).

[8] Tsishkou,D., Hammami, M. and Chen, L.: Face Detection in Video Using Combined Data-mining and Histogram based Skin-color Model, IEEE Third International Symposium on Image and Signal Processing and Analysis, Rome, Italy (2003) 500-503.

[9] Hammami, M., Tsishkou, D. and Chen, L.: Data-mining based Skin-color Modeling and Applications, Third International Workshop on Content-Based Multimedia Indexing, Rennes, France (2003) 157-162.

[10] Terrillon, J.C., Shirazi, M., Fumakachi, H. and Akamatsu, S.: Comparative performance of different skin chrominance models and chrominances spaces for the automatic detection of human faces in color images, in Proceedings of IEEE International Conference on Face and Gesture Recognition, Grenoble (200) 54-61.

[11] Brunelli, R. and Poggio, T.: Face recognition: features versus templates, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 10 (1993) 1042-1052.

[12] Bellaroussi, R. : localisation du visage dans des images et sequences video couleur. University of Pierre et Marie Curie. Thèse de doctorat (2006).

[13] Aznaveh, M.M., Mirzaei, H., Roshan, E. and Saraee, M.: A new color based method for skin detection using RGB vector space, Human System Interactions Conference (2008) 932-935.

[14] Aznaveh, M.M., Mirzaei, H., Roshan, E. and Saraee, M.: A New and Improved Skin Detection Method Using Mixed Color Space, Human System Interactions Conference, Vol. 60 (2009) 471-480.

[15] Nguyen, T.P.: Détection de visage dans images de couleur en utilisant une caractéristique invariante. IEEE International Conference on Sciences of Electronic, Technologies of Information and Telecommunications SETIT, Sousa, Tunisie (2005)

[16] Korosee, J., Gyergyek, L., Paversie, N.: Face contour detection based on the Hough Transform. Automatika,Vol. 31 (1990).

[17] Fan, L. and Sung, K.-K.: Model-based varying pose face detection and facial feature registration in colour images. Pattern Recognition Letters, Vol. 24, No. 1-3 (2003) 237-249.

[18] Augusteijn, M.F. and Skufca, T.L.: Identification of human faces through texture-based feature recognition and neural network technology, IEEE International Conference on Neural Networks, Vol. 1 (1993) 392-398.

[19] Hsi-Jian, L. and Yu-Cheng, T.: Measurement of Face Recognizability for Visual Surveillance. Pattern Recognition and image Analysis. LNCS, Vol. 3687 (2005) 287-294.

[20] Bray, T.: Measuring the Web. Proceedings for the 5th International World Wide Web Conference, France (1996).

[21] Coffman, K.G. and Odlyzko, A.: The Size and Growth Rate of the Internet. Technical Report, Vol. 21 (1999) 99-11.

[22] Cavanagh, J.P.: Frame Relay Applications: Business and Technical Case Studies. San Francisco: Morgan Kaufman, (1998).

[23] Mittal, A.: An Overview of Multimedia Content-Based Retrieval Strategies. Informatica, Vol. 3 (2006) 347-356.

[24] Chang, N.S. and Fu, K.S.: Query-by-Pictorial-Example. IEEE Transactions on Software Engineering, Vol. 6 (1980) 519-524.

[25] Chang, S.-K. and Kunii, T. : Pictorial database systems. IEEE Computer Magazine, Vol. 14 (1981) 13-21.

[26] Dimitroff, D.C., Chang, S.K., Yan, C.W. and Arndt, T.: An Intelligent Image Database System. IEEE Transactions on Software Engineering, Vol. 14, No. 5 (1988) 681-688.

[27] Sigal, L., Sclaroff, S. and Athitsos, V.: Estimation and Prediction of Evolving Color Distributions for Skin Segmentation under Varying Illumination, Proceedings of IEEE Conf. Computer Vision and Pattern Recognition, vol. 2 (2000) 152-159.
[28] Gevers, T. and Kajcovski, V.K.: Image segmentation by directed region subdivision. 12 th International Conference on Pattern Recognition, Vol. 1, Conference A, Jerusalem, Israel (1994) 342-346.
[29] Li, X.Q., Zhao, Z.W., Cheng, H.D., Huang, C.M. and Harris, R.W.: A fuzzy logic approach to image segmentation. 12th International Conference on Pattern Recognition, Vol. 1, Conference A (1994) 337-341.
[30] Tamura, H., Mori, T. and Yamawaki, T.: Textural features corresponding to visual perception. IEEE Systems, Man, and Cybernetics, Vol. 8 (1978) 460-473.
[31] Hafner, W., and Munkelt, O.: Using Color for Detecting Persons in Image Sequences, Pattern Recognition and Image Analysis, Vol. 7, No. 1 (1997) 47-52.
[32] Jones, M.J. and Rehg, J.M.: Statistical Color Models with Application to Skin Detection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1 (1999) 274-280.
[33] Keren, D., Osadchy, M. and Gotsman, C.: Antifaces: A novel, fast method for image detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 23, No. 7 (2001) 747-761.
[34] Barber, R., Niblack, W., Flickner, M., Glasman, E., Petkovic, D. Yanker, P., Faloutsos, C. and Taubin, G.: The QBIC project: Querying Images by Content Using Color, Texture, and Shape. In Proceedings of the IS&T SPIE Storage and Retrieval for Image and Video Databases, San Jose, CA, USA (1993) 173-187.
[35] Rui, Y., Huang, T. and Chang, S.: Image Retrieval: Current Techniques, Promising Directions and Open Issues. Journal of Visual Communication and Image Representation, Vol. 10, No. 4 (1999)39-62.
[36] Rui, Y., Huang, T.S. and Mehrotra, S.: Content-based Image Retrieval with Relevance Feedback in MARS. In Proceedings of the IEEE International Conference on Image Processing, ICIP ’97, Santa Barbara, California, USA (1997) 815-818.
[37] Kompatsiaris, I., Triantafillou, E. and Strintzis, M.G.: Region-Based Color Image Indexing and Retrieval. In Proceedings of the International Conference on Image Processing (ICIP2001), Vol. 1, Thessaloniki, Greece (2001) 658-661.
[38] Pass, G., Zabih, R. and Miller, J.: Comparing images using color coherence vectors. In Proceedings of the Fourth ACM Multimedia Conference, New York (1996) 65-74.
[39] Swain, M.J. and Ballard, D.H.: Color indexing. International Journal of Computer Vision, Vol. 7, No. 1 (1991) 11-32.
[40] Smith, J.R. and Chang, S.-F.: Querying by color regions using the VisualSEEk content-based visual query system. Intelligent Multimedia Information Retrieval, MIT Press (1997) 159-173.
[41] Witten, I.H. and Frank, E.: Data mining: Practical machine learning tools and techniques, Morgan Kaufmann (2005).
[42] Pyle, D.: Business Modeling and Data Mining. Morgan Kaufmann (2003).
[43] Awad, M., Khan, L., Thuraisingham, B. and Wang, L.: Design and Implementation of data mining tools. Auerbach Publications (2009).
[44] Huang, T.-M., Kecman, V. and Kopriva, I.: Kernel Based Algorithms For Mining Huge Data Sets: Supervised, Semi-supervised, and Unsupervised Learning. Springer-Verlag (2006).
[45] Vercellis, C.: Business Intelligence: Data Mining and Optimization for Decision Making, Wiley (2009).