Video Signals

19 Jan 2021

You are working for a TV program and you are asked to develop a script able to automatically detect people faces inside of an image and pixelate them in order to protect their identity. Although this could be achieved using many different methods, you decided to detected faces using color segmentation and morphological operations. The main idea is to use segmentation in the HSV color space in order to detect skin regions inside the image and then use morphological operations in order to detect face shaped areas. In order to perform the first task (skin segmentation) you need to estimate how the skin colors are distributed inside the HSV color space. In order to obtain this information, you can use a small database of 100 32x32 color face images that can be extracted from the compressed file *facesdb.zip*. Extract the zip file and build a simple MATLAB script able to perform the following steps:

a) Cycle through all the jpg images inside the facesdb folder (*hint:* use `sprintf('facesdb/%03d.jpg',i)` in order to obtain each filename, where *i* is the cycle variable)
b) For each image convert it into HSV color space, calculate the median hue and saturation values and store them inside two vectors
c) Outside of the cycle calculate the average and standard deviation for hue and saturation median vectors, obtaining μ_H , σ_H and μ_S , σ_S for Hue and Saturation respectively.

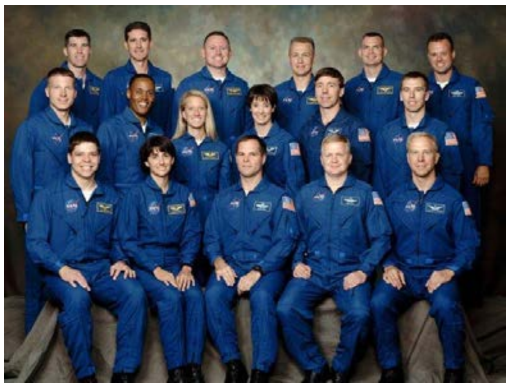
Figure 1: input image

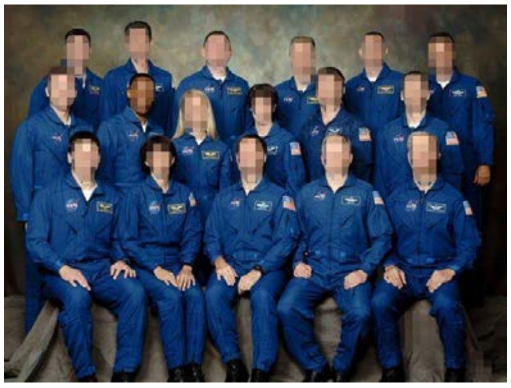
Figure 2: output image

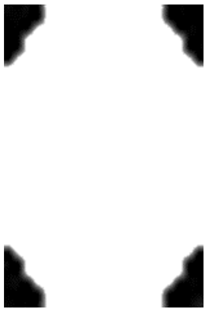
Figure 3: structuring element

After obtaining these skin color information, write a script able to, starting from Fig.1 produce the result reported in Fig.2. In particulare, write a MATLAB script able to perform the following steps:

a) Read the 8-bit input color image (stored in the file *'input_image.jpg'*) and visualize it.
b) Obtain the HSV representation of the input image and build a binary mask image using the following conditions: $\mu_H - 1.5\cdot\sigma_H < H < \mu_H + 1.5\cdot\sigma_H$ and $\mu_S - 1.5\cdot\sigma_S < S < \mu_S + 1.5\cdot\sigma_S$
c) Using a square structuring element of size 15 px, apply a morphological operation to the obtained mask in order to remove small isolated black regions.
d) Using a 30x20 structuring element depicted in Fig.3 apply a morphological opening in order to keep in the mask just regions related to faces (*hint:* start from a 10 radius disk structuring element and stretch it in order to obtain the desired shape).
e) Obtain a pixeled version of the image by reducing its dimension by 1/10 and expanding it back to its original size using an interpolation method that could achieve the desired output.
f) Obtain the output image by starting from the original image and substituting it with the pixeled one just in the regions in which the mask is set to TRUE. Visualize the result image.

<u>Name your solution file as: *name_surname_*20210191.m</u>

Solution

```
clc
close all
clear all
n = 100;
%1.a)
for i=1:n
    I = imread(sprintf('facesdb/%03d.jpg',i));
    %1.b)
    I_hsv = rgb2hsv(I);
    I_h = I_hsv(:,:,1);
    I_s = I_hsv(:,:,2);
    avg_h(i) = median(I_h(:));
    avg_s(i) = median(I_s(:));
end

%1.c)
mean_h=mean(avg_h);
mean_s=mean(avg_s);
std_h=std(avg_h);
std_s=std(avg_s);

%2.a)
I = imread('input_image.jpg');
imshow(I)

%2.b)
I_hsv = rgb2hsv(I);
mask = I_hsv(:,:,1)>mean_h-1.5*std_h & I_hsv(:,:,1)<mean_h+1.5*std_h & ...
       I_hsv(:,:,2)>mean_s-1.5*std_s & I_hsv(:,:,2)<mean_s+1.5*std_s;

%2.c)
SE_1 = strel('square',15);
mask = imclose(mask,SE_1);

%2.d)
SE_2 = strel('disk',10);
A = imresize(SE_2.Neighborhood,[30,20]);
SE_2 = strel('arbitrary',A);
mask = imopen(mask,SE_2);

%2.e)
face_pixeled = imresize(I, 0.1);
face_pixeled = imresize(face_pixeled, [size(I,1) size(I,2)], 'nearest');

%2.f)
face_out = face_pixeled.*uint8(mask) + I.*uint8(1-mask);
figure
imshow(face_out)
```