

## Color Image Processing and Skin Detection

Seyed Hesamoddin Hosseini

### خلاصه

در بخش اول این میانی پروژه سه فضای رنگ به جز فضاهای رنگ معرفی شده در درس را معرفی کرده و در هر مورد ویژگی های فضای رنگ، کاربردها و ارتباط آن با فضای رنگ RGB یا HSI صحبت نموده ایم.

در بخش دوم، یکی از الگوریتم های ساده و کارا برای تشخیص پوست (Skin Detection)، که الگوریتمی مبتنی بر تمایز رنگ است را مورد بررسی قرار خواهیم داد. این الگوریتم به نام پس افکنش هیستوگرام (backprojection histogram) شناخته می شود.

در این الگوریتم ابتدا تصویر را به فضای رنگ تبدیل می کنیم که در آن کروماتیسیتی استخراج شود. در اینجا برای این کار از فضای رنگ نرمالیزه شده استفاده می کنیم. برای این منظور فضای رنگ RGB را نرمالیزه می کنیم. در تشخیص رنگ پوست، تنها از سطح های قرمز و سبز استفاده می کنیم و از سطح آبی صرف نظر می کنیم. حال در این تصویر میزان رنگ سبز و قرمز هر پیکسل را بدست می آوریم و هیستوگرام دو بعدی را تشکیل می دهیم و سپس تصویر خاصی که می خواهیم پوست را در آن تشکیل بدھیم را روی هیستوگرام اعمال می کنیم تا پوست را بدست آوریم. برای این منظور، مقادیر  $r$  و  $g$  در این بخش از تصویر را به  $N$  سطح کوانتايز می کنیم. ماتریس  $H$  به ابعاد  $N \times N$  با مقادیر صفر را تشکیل می دهیم و سپس تصویر را اسکن کرده و پوست را تشخیص می دهیم.

برای پیاده سازی این پروژه از Matlab استفاده می نماییم.

### شرح تکنیکال:

#### مرحله اول:

با توجه به توضیحات داده شده در بخش اول این میانی پروژه سه فضای رنگ معرفی شده در درس را معرفی کرده و در هر مورد ویژگی های فضای رنگ، کاربردها و ارتباط آن با فضای رنگ RGB یا HSI صحبت نموده ایم.

### فضاهایی رنگی YIQ و YU

این فضاهای با تغییرات کوچکی از روی مدل  $R'G'B'$  بدست آمده است. این تغییرات برای آن است که مدل جدید برای انتقال، نسبت به مدل RGB کارآمدتر شود و همین طور با سیستم تلویزیون های سیاه و سفید، سازگار گردد. در حقیقت جزء  $Y$  از این فضاهای اطلاعات ویدیویی مورد نیاز یک سیستم تلویزیون سیاه و سفید را بطور کامل تامین می کند. برای تبدیل فضای RGB به فضای YIQ و YUV و بالعکس از روابط زیر استفاده می شود:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \quad \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 1 & 0.9557 & 0.6199 \\ 1 & -0.2716 & -0.6469 \\ 1 & -1.1082 & 1.7051 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix}$$

تبدیل از فضای رنگ RGB به فضای YIQ (سیستم PAL) و بالعکس

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} \quad \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.1398 \\ 1 & -0.3946 & -0.5805 \\ 1 & 2.0320 & -0.0005 \end{bmatrix} \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

تبدیل از فضای رنگ RGB به فضای YUV (NTSC) و بالعکس

### مدل رنگ \*L\*a\*b\*

فضای رنگ  $L^*a^*b^*$  از نظر ادراکی یکنواخت هستند. مقادیر  $L^*a^*b^*$  از طریق فرمولهای زیر محاسبه می شود:

$$L^* = \begin{cases} 116\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}} - 16 & \text{if } \frac{Y}{Y_n} > 0.008856 \\ 903.3\left(\frac{Y}{Y_n}\right) - 16 & \text{if } \frac{Y}{Y_n} \leq 0.008856 \end{cases}$$

$$a^* = 500 \left[ f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right]$$

$$b^* = 200 \left[ f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right]$$

مقادیر Z، X و Y براساس درصد رنگهای اصلی به صورت زیر محاسبه می‌شوند:

$$X = 0.607R + 0.174G + 0.201B$$

$$Y = 0.299R + 0.587G + 0.114B$$

$$Z = 0.000R + 0.066G + 1.117B$$

افراد تصور می‌کنند که افراد مختلف دارای رنگهای مختلف پوست هستند، اما مطالعات و تحقیقات انجام گرفته نشان می‌دهد که تغییر رنگ بیشتر در اثر تغییر غلظت رنگ در پوست افراد می‌باشد و با اندازه گیری دستهای افراد مختلف در شرایط نوری متفاوت، تغییرات اندکی در مقادیر a\*، b\* و L\* مشاهده شده است.

## مدل رنگ Lab

یک مدل تخیلی از حیطه رنگ‌های قابل رویت توسط انسان است. در این مدل L مخفف عبارت Lightness به معنای درخشندگی است، a و b نیز برچسب‌های محورهای در بردارنده مدل هستند. مدل رنگ Lab مدل جامعی بوده و مدل رنگ CMYK و RGB زیر مجموعه‌ای از این مدل هستند. برخلاف مدل‌های رنگی RGB و CMYK، رنگ‌های Lab، تقریباً نزدیک به بینایی انسان طراحی شده است.

برخی از کاربردهای مدل رنگ LAB در نرم افزارهای خاص عبارتست از:

-در نرم افزار Adobe Photoshop، به منظور ویرایش تصویر از LAB استفاده می‌شود.

-در پروفایلهای ICC، فضای رنگ Lab اکه به عنوان یک ارتباط دهنده پروفایل به کار می‌رود.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.4124 & 0.3576 & 0.1805 \\ 0.2126 & 0.7152 & 0.0722 \\ 0.0193 & 0.1192 & 0.9505 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 0.3897 & 0.6890 & -0.0787 \\ -0.2298 & 1.1834 & 0.0464 \\ 0.0000 & 0.0000 & 1.0000 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

$$\begin{bmatrix} L \\ \alpha \\ \beta \end{bmatrix} = \text{diag} \left( \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{6}}, \frac{1}{\sqrt{2}} \right) \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} \log L \\ \log M \\ \log S \end{bmatrix}$$

مرحله دوم:

با توجه به توضیحات داده شده در بخش دوم، یکی از الگوریتم‌های ساده و کارا برای تشخیص پوست (Skin Detection)، که الگوریتمی مبتنی بر تمایز رنگ است را مورد بررسی قرار خواهیم داد. این الگوریتم به نام پس‌افکنش هیستوگرام (backprojection histogram) شناخته می‌شود.

در این الگوریتم ابتدا تصویر را به فضای رنگ تبدیل می‌کنیم که در آن کروماتیسیتی استخراج شود. در اینجا برای این کار از فضای رنگ نرمالیزه شده استفاده می‌کنیم. برای این منظور فضای رنگ RGB را نرمالیزه می‌کنیم.



شکل ۱ - تصویر پوست های مختلف

در تشخیص رنگ پوست، تنها از سطح های قرمز و سبز استفاده می کنیم و از سطح آبی صرف نظر می کنیم.

```
I = image (:,:,1) + image (:,:,2) + image (:,:,3);
R = image (:,:,1). / I;
G = image (:,:,2). / I;
B = image (:,:,3). / I;
```

حال در این تصویر میزان رنگ سبز و قرمز هر پیکسل را بدست می آوریم و هیستوگرام دو بعدی را تشکیل می دهیم و سپس تصویر خاصی که می خواهیم پوست را در آن تشکیل بدهیم را روی هیستوگرام اعمال می کنیم تا پوست را بدست آوریم. برای این منظور، مقادیر  $r$  و  $g$  در این بخش از تصویر را به  $N$  سطح کوانتایز می کنیم. برای این منظور مقادیر سطح قرمز و سبز را به 128 (یا هر تعداد دیگر) سطح تقسیم می کنیم. پس تک تک مقادیر سبز و قرمز را در 128 ضرب کرده و آن را به بالا یا پایین رند می کنیم.

در اینجا مقدار  $N$  را برابر 128 در نظر گرفته ایم.

```
nr = zeros(img_row_size, img_col_size);
ng = zeros(img_row_size, img_col_size);

for row = 1:img_row_size
    for col = 1:img_col_size

        nr(row, col) = round(R(row, col) * N);
        ng(row, col) = round(G(row, col) * N);
    end
end
```

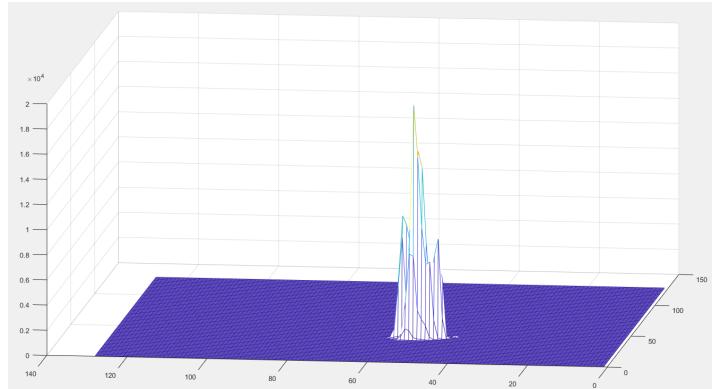
ماتریس  $H$  به ابعاد  $N \times N$  با مقادیر صفر را تشکیل می دهیم. حال به ترتیب مقادیر سبز و قرمز هر پیکسل تصویر پوست، که به 128 سطح تقسیم شده است، را می خوانیم. مقدار سطح قرمز، شماره سطر و مقدار سطح سبز شماره‌ی ستون را به ما می دهد. در هیستوگرام مقدار این سطر و ستون را یک واحد اضافه می کنیم. این کار را تا انتها انجام می دهیم تا هیستوگرام تکمیل شود.

```
H = zeros(N, N);

for row = 1:img_row_size
    for col = 1:img_col_size

        r = nr(row, col);
        c = ng(row, col);

        H(r, c) = H(r, c) + 1;
    end
end
```



شکل 2- هیستوگرام دو بعدی

حال یک تصویر را از ورودی دریافت می کنیم و مانند تصویر پوست سطوح سبز و قرمز آن را نرمالیزه می کنیم و به 128 سطح تقسیم بندی می کنیم. حال به ازای مقدار سطح هر پیکسل در صفحه قرمز و سبز، به یک سطر و ستون خاص از هیستوگرام دو بعدی H می رویم و مقدار آن را برای همان پیکسل در تصویر خروجی قرار می دهیم.

```

for row = 1:img_row_size
    for col = 1:img_col_size

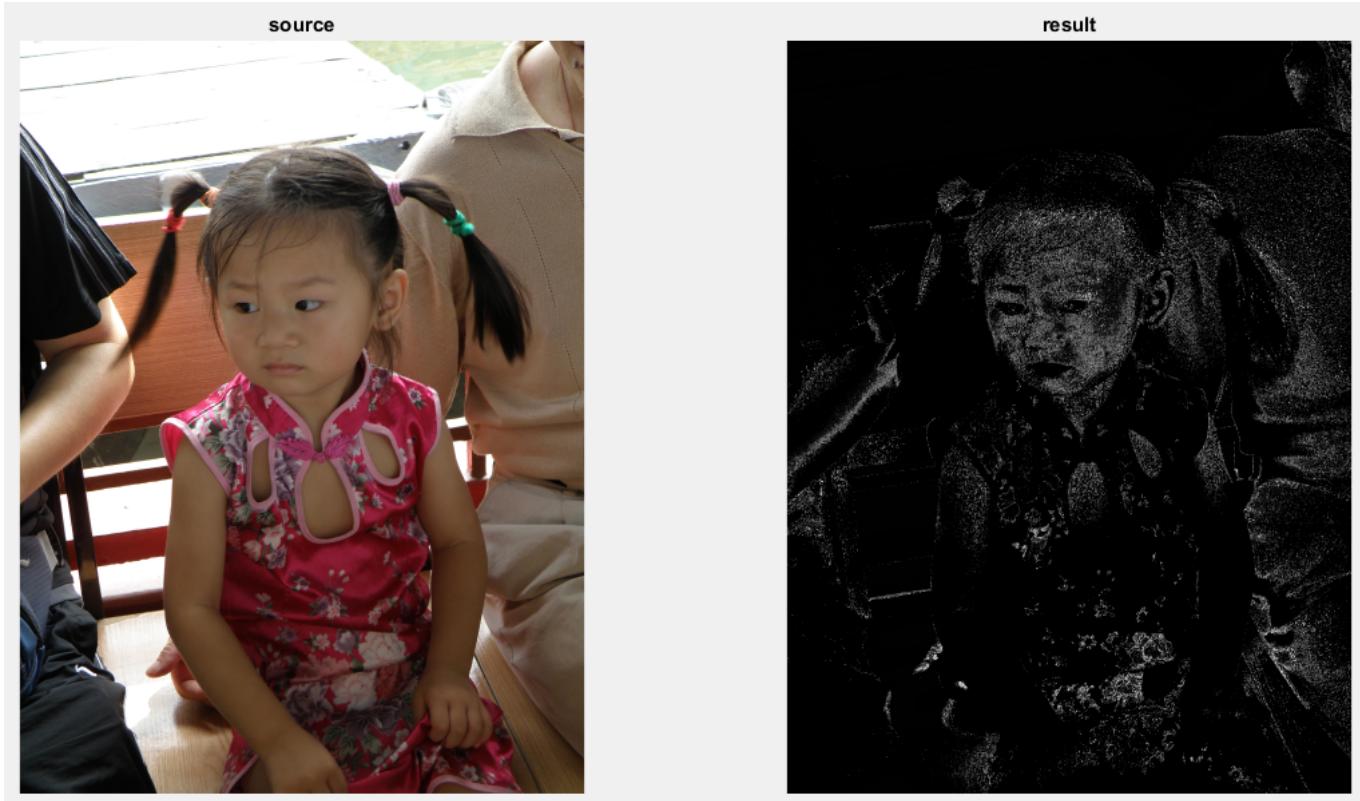
        r = uint8(nr(row, col));
        c = uint8(nc(row, col));

        if r == 0
            r = 1;
        end
        if c == 0
            c = 1;
        end

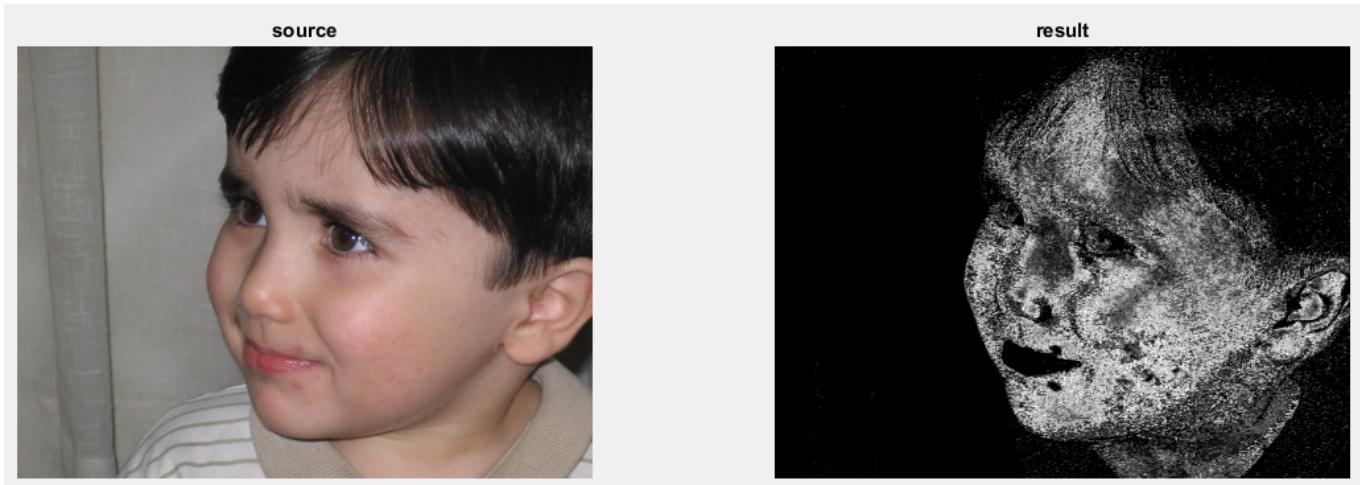
        Likelihood_1911(row, col) = H(r, c);
    end
end

```

در ادامه خروجی این الگوریتم را برای ورودی های تصویر مختلف، مشاهده می نمایید:



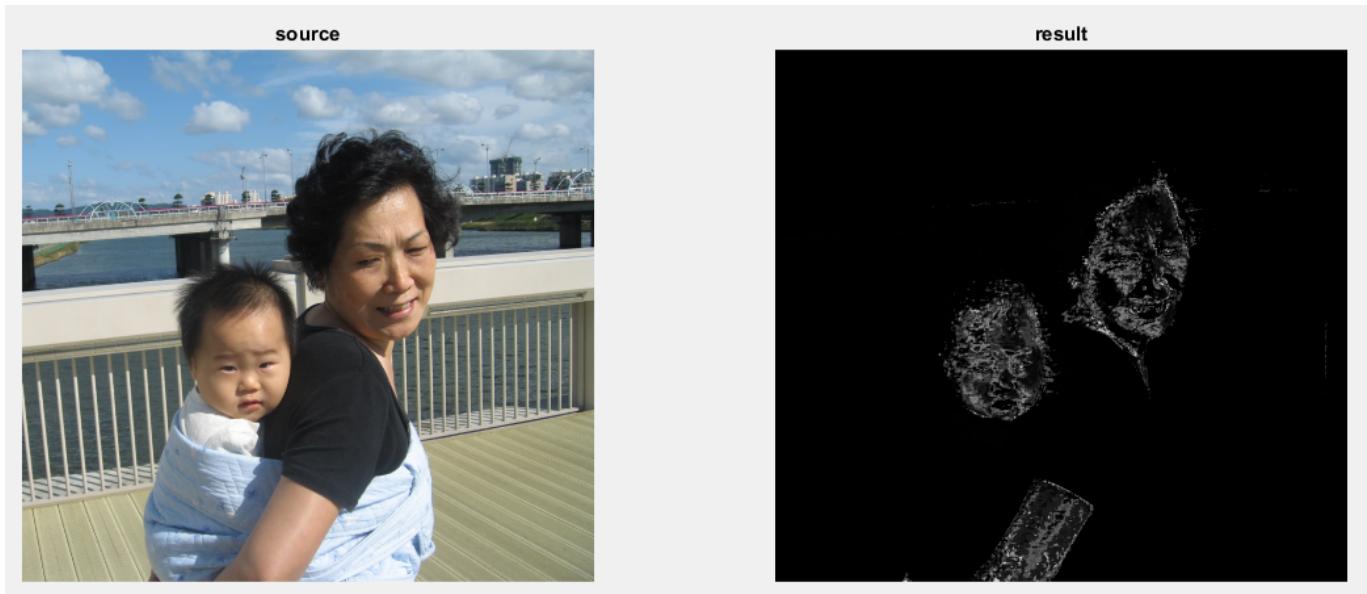
شكل 3 - تصوير اصلی و شناسایی پوست توسط الگوریتم



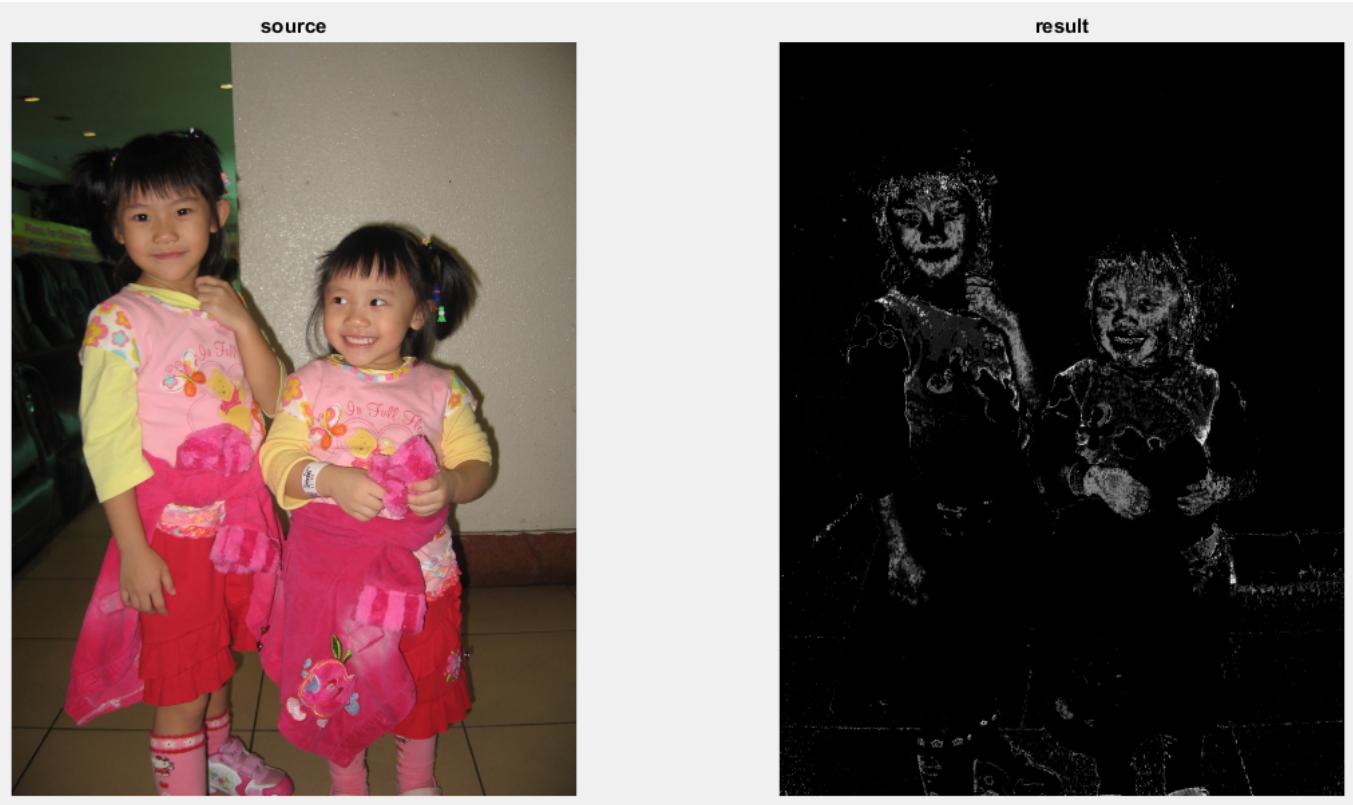
شكل 4 - تصوير اصلی و شناسایی پوست توسط الگوریتم



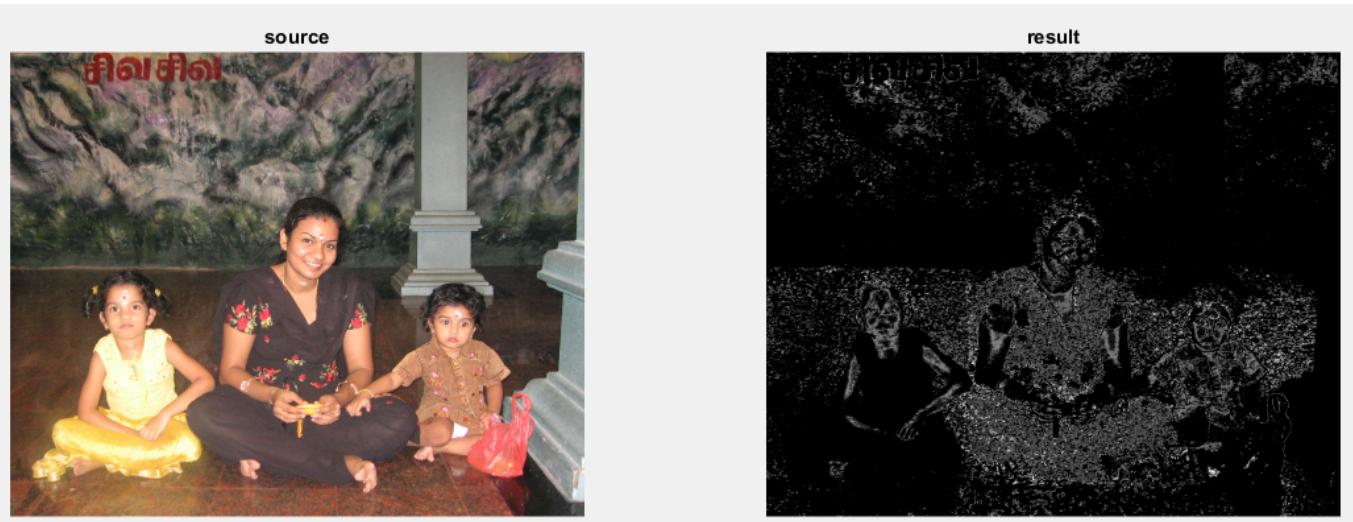
شكل 5 - تصوير اصلي و شناسابي پوست توسط الگوريتم



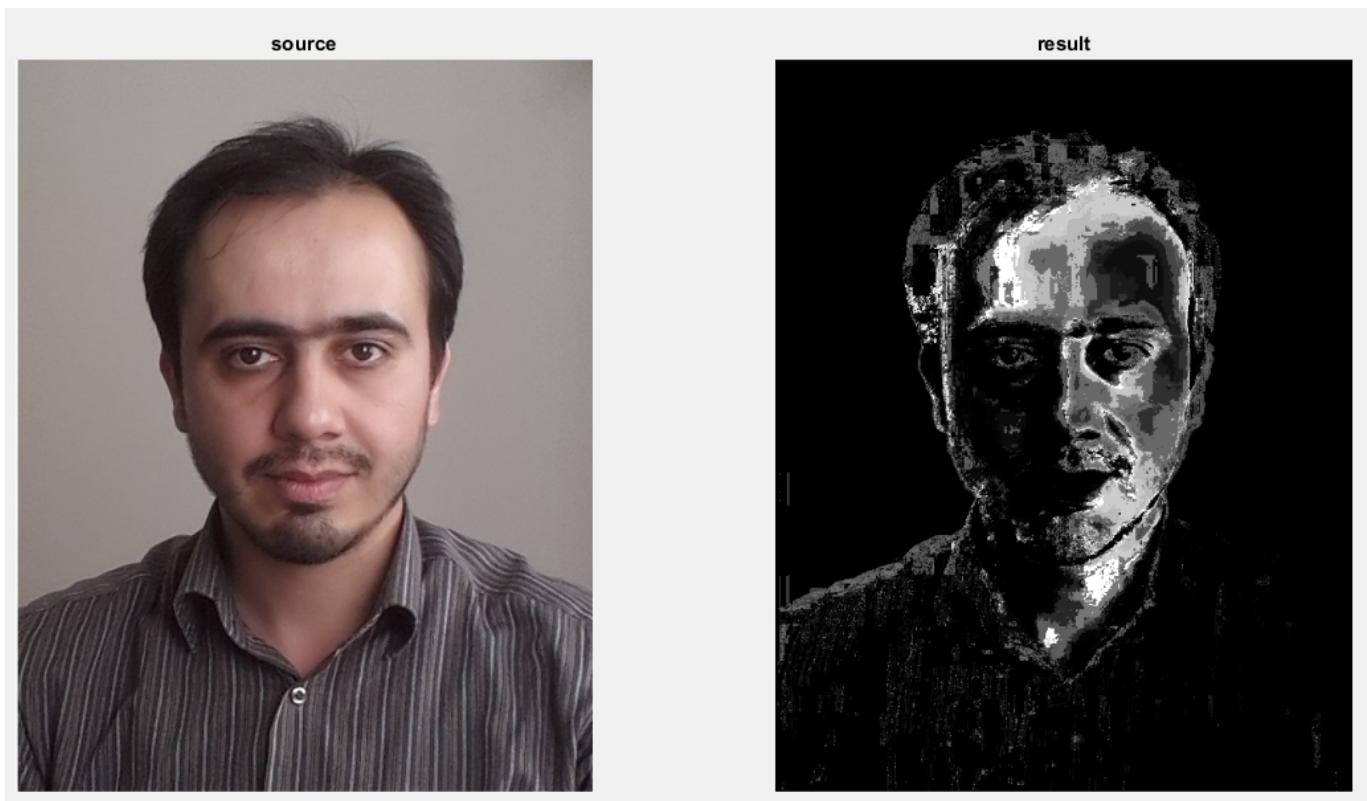
شكل 6 - تصوير اصلي و شناسابي پوست توسط الگوريتم



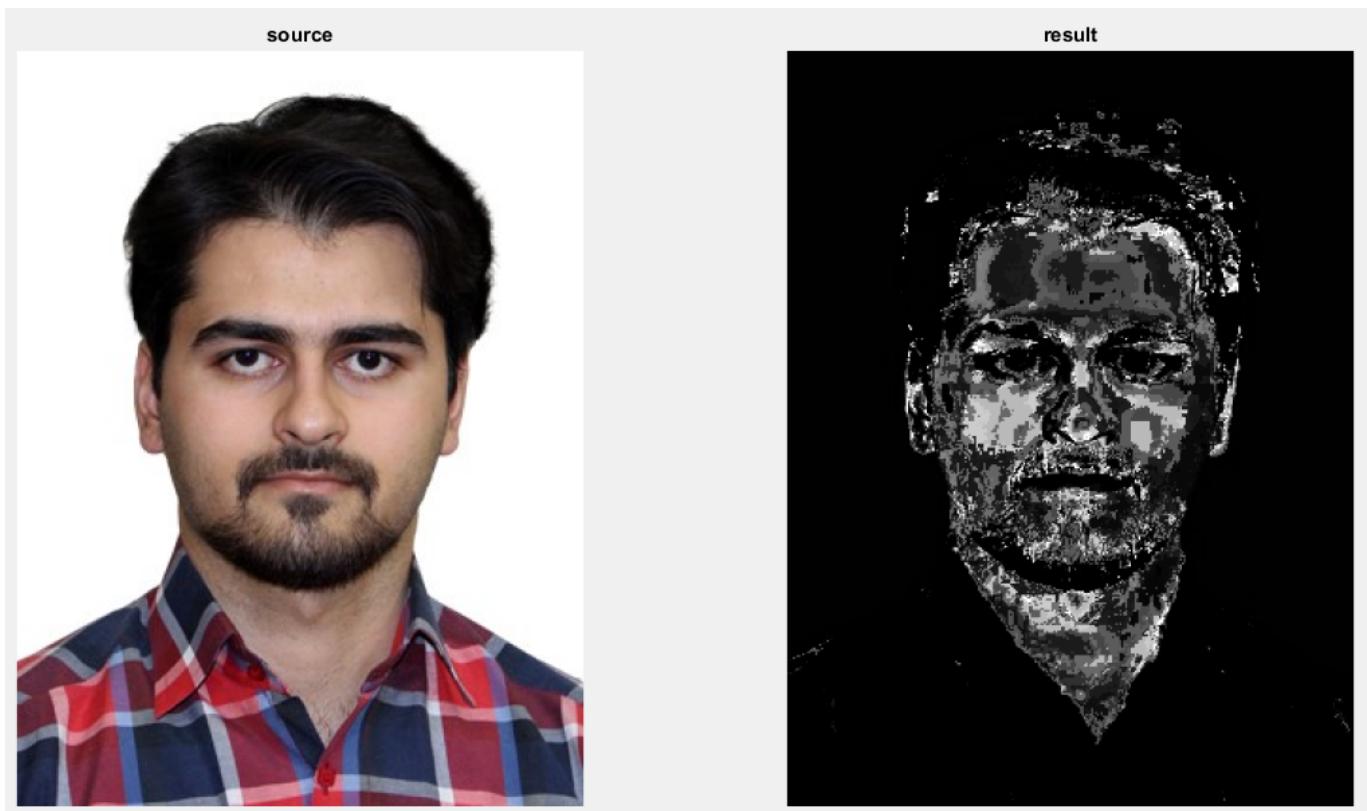
شكل 7 - تصوير اصلي و شناسابي پوست توسط الگوريتم



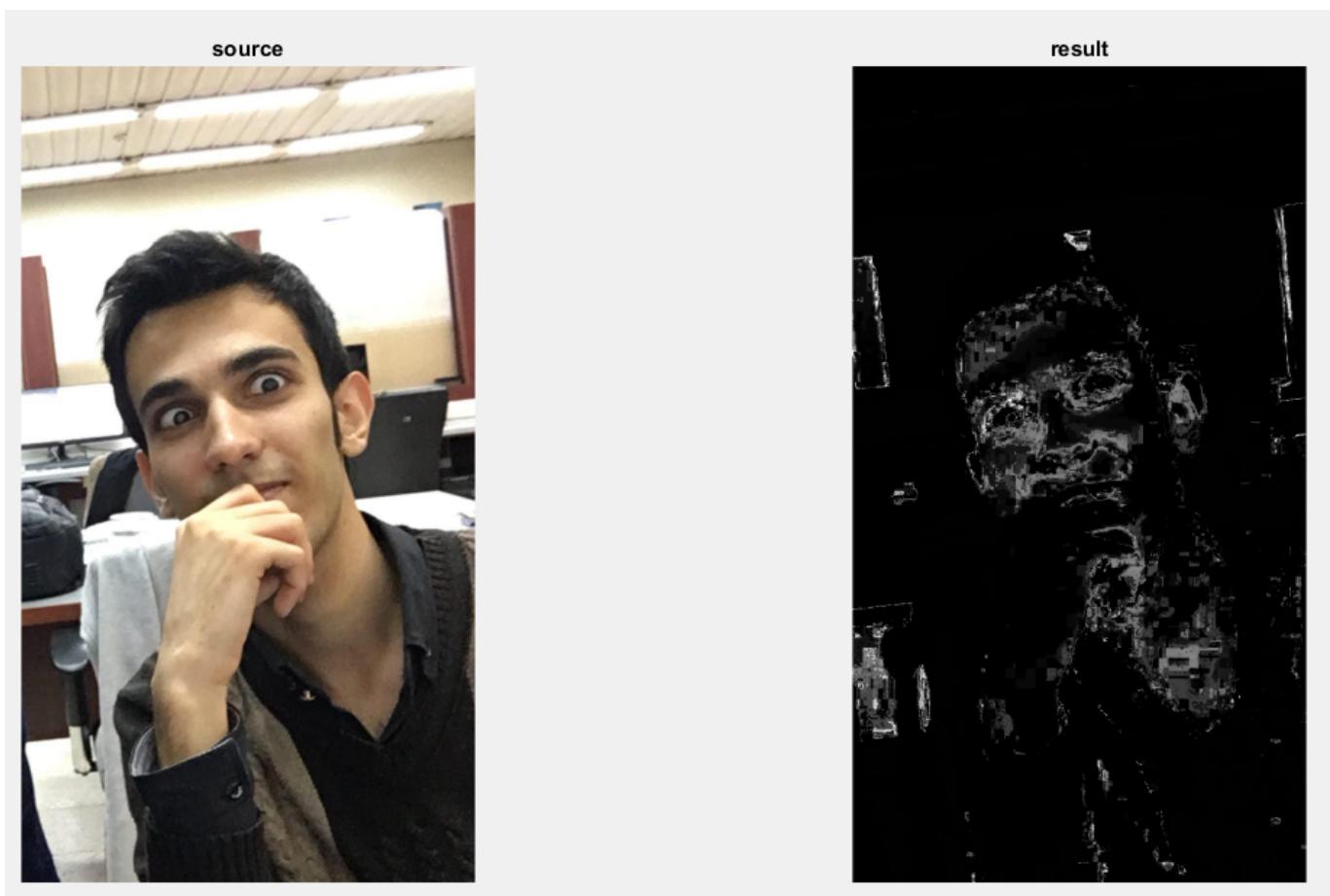
شكل 8 - تصوير اصلي و شناسابي پوست توسط الگوريتم



شکل ۹- تصویر اصلی و شناسایی پوست توسط الگوریتم



شکل ۱۰ - تصویر اصلی و شناسایی پوست توسط الگوریتم



شکل 11 - تصویر اصلی و شناسایی پوست توسط الگوریتم

**نتیجه گیری:**

در این بخش میخواهیم به این سوالات پاسخ دهیم:

چرا الگوریتم تمام نقاط پوست را تشخیص نداده و پیکسل های را نیز به غلط پوست تشخیص می دهد؟

دلیل اینکه برخی پیکسل ها به غلط، مانند پوست تشخیص داده می شوند، شبیه بودن رنگ آن نقاط به رنگ پوست می باشد. مانند پیراهن فرد سمت راست در شکل 3

و دلیل اینکه برخی نقاط پوست، تشخیص داده نمی شوند، به دلیل این است که شدت روشنایی آنها تنظیم نمی باشد. و از آنجا که فضای رنگ RGB نمی تواند بین رنگ و شدت روشنایی، تمایزی قائل شود، این نقاط به عنوان یک رنگ دیگر، شناخته می شوند و در نتیجه به عنوان پوست تشخیص داده نمی شوند.

**پیوست:**

سورس کد پروژه با نرم افزار متلب:

پیاده سازی پروژه:

```
close all;
clear;

N = 128;
image=imread('D:/Im1910.png');
image = im2double(image);
[img_row_size, img_col_size] = size(image(:,:,1));

I = image(:,:,:,1) + image(:,:,:,2) + image(:,:,:,3);
R = image(:,:,:,1)./I;
G = image(:,:,:,2)./I;
```

```

B = image (:,:,3) ./I;

nr = zeros(img_row_size, img_col_size);
ng = zeros(img_row_size, img_col_size);

for row = 1:img_row_size
    for col = 1:img_col_size

        nr(row, col) = round(R(row, col) * N);
        ng(row, col) = round(G(row, col) * N);
    end
end

H = zeros(N, N);

for row = 1:img_row_size
    for col = 1:img_col_size

        r = nr(row, col);
        c = ng(row, col);

        H(r, c) = H(r, c) + 1;
    end
end

Img1911 = imread('D:/Im1914.jpg');
Img1911 = im2double(Img1911);

[img_row_size, img_col_size] = size(Img1911(:,:,1));
I = Img1911(:,:,1) + Img1911(:,:,2) + Img1911(:,:,3);
R = Img1911(:,:,1) ./ I;
G = Img1911(:,:,2) ./ I;
B = Img1911(:,:,3) ./ I;

nr = zeros(img_row_size, img_col_size);
ng = zeros(img_row_size, img_col_size);

for row = 1:img_row_size
    for col = 1:img_col_size

        nr(row, col) = round(R(row, col) * N);
        ng(row, col) = round(G(row, col) * N);
    end
end

Likelihood_1911 = zeros(img_row_size, img_col_size);

for row = 1:img_row_size
    for col = 1:img_col_size

        r = uint8(nr(row, col));
        c = uint8(ng(row, col));

        if r == 0
            r = 1;
        end
        if c == 0
            c = 1;
        end

        Likelihood_1911(row, col) = H(r, c);
    end
end

```

```
figure;
subplot(1,2,1);
imshow(Img1911,[])
title('source');
subplot(1,2,2);
imshow(Likelihood_1911,[])
title('result');
```