```
sample1 = double(imread('training_A.bmp'));

patch1 = (sample1(122:150, 297:338, :));
patch2 = (sample1(246:265, 117:145, :));

red1 = patch1(:,:,1);
green1 = patch1(:,:,2);
blue1 = patch1(:,:,3);

ss1_red2 = patch2(:,:,1);
ss1_green2 = patch2(:,:,3);
ss1_blue2 = patch2(:,:,2);

red1 = red1(:);
green1 = green1(:);
blue1 = blue1(:);

ss1_red2 = ss1_red2(:);
ss1_green2 = ss1_green2(:);
ss1_blue2 = ss1_blue2(:);

[Xs, Ys] = size(red1);
temp_red = zeros(2, Ys);
temp_red(1, 1:length(red1)) = red1;
temp_red(2, 1:length(ss1_red2)) = ss1_red2;

total_red = temp_red(:);

[g_rows, g_cols] = size(green1);
temp_green = zeros(2, g_cols);
temp_green(1, 1:length(green1)) = green1;
temp_green(2, 1:length(ss1_green2)) = ss1_green2;

total_green = temp_green(:);

[b_rows, b_cols] = size(blue1);
temp_blue = zeros(2, b_cols);
temp_blue(1, 1:length(blue1)) = blue1;
temp_blue(2, 1:length(ss1_blue2)) = ss1_blue2;

total_blue = temp_blue(:);

total_color = total_red + total_green + total_blue;
red_norm = total_red ./ total_color;
red_norm(isnan(red_norm)) = 0;

green_norm = total_green ./ total_color;
green_norm(isnan(green_norm)) = 0;

red_mean = mean(red_norm);
green_mean = mean(green_norm);
red_std = std(red_norm);
green_std = std(green_norm);

read = double(imread('test.bmp'));

[Xs, Ys, Zs] = size(read);
skin_detection1 = zeros(Xs, Ys);
```

```
for x = 1:Xs
    for y = 1:Ys
        red = read(x, y, 1);
        green = read(x, y, 3);
        blue = read(x, y, 2);

        sum = red+green+blue;
        if sum > 0
            main_red = red / sum;
            main_green = green / sum;
        else
            main_red = 0;
            main_green = 0;
        end

        red_prob = gaussian_probability(red_mean, red_std, main_red);
        green_prob = gaussian_probability(green_mean, green_std, main_green);
        prob = red_prob * green_prob;
        skin_detection1(x, y) = prob;
    end
end

%%%Module 2:


sample2 = double(imread('training_B.bmp'));
ss2_patch1 = (sample2(189:230, 213:293, :));
ss2_patch2 = (sample2(376:423, 63:120, :));
ss2_patch3 = (sample2(413:471, 404:455, :));

ss2_red = ss2_patch1(:,:,1);
ss2_green = ss2_patch1(:,:,2);
ss2_blue = ss2_patch1(:,:,3);

ss2_red2 = ss2_patch2(:,:,1);
ss2_green2 = ss2_patch2(:,:,2);
ss2_blue2 = ss2_patch2(:,:,3);

ss2_red3 = ss2_patch3(:,:,1);
ss2_green3 = ss2_patch3(:,:,2);
ss2_blue3 = ss2_patch3(:,:,3);

ss2_red = ss2_red(:);
ss2_green = ss2_green(:);
ss2_blue = ss2_blue(:);

ss2_red2 = ss2_red2(:);
ss2_green2 = ss2_green2(:);
ss2_blue2 = ss2_blue2(:);

ss2_red3 = ss2_red3(:);
ss2_green3 = ss2_green3(:);
ss2_blue3 = ss2_blue3(:);

[Xs, s2p1Ys] = size(ss2_red);
red_temp2 = zeros(3, s2p1Ys);
red_temp2(1, 1:length(ss2_red)) = ss2_red;
red_temp2(2, 1:length(ss2_red2)) = ss2_red2;
```

```
red_temp2(3, 1:length(ss2_red3)) = ss2_red3;

total_Red2 = red_temp2(:);


[Xs, s2p1_greenYs] = size(ss2_green);
green_temp2 = zeros(3, s2p1_greenYs);
green_temp2(1, 1:length(ss2_green)) = ss2_green;
green_temp2(2, 1:length(ss2_green2)) = ss2_green2;
green_temp2(3, 1:length(ss2_green3)) = ss2_green3;

totalGreen2 = green_temp2(:);

[Xs, s2_blueYs] = size(ss2_blue);
blue_temp2 = zeros(3, s2_blueYs);
blue_temp2(1, 1:length(ss2_blue)) = ss2_blue;
blue_temp2(1, 1:length(ss2_blue2)) = ss2_blue2;
blue_temp2(1, 1:length(ss2_blue3)) = ss2_blue3;

totalBlue2 = blue_temp2(:);

total_color2 = total_Red2 + totalGreen2 + totalBlue2;
red_norm2 = total_Red2 ./ total_color2;
red_norm2(isnan(red_norm2)) = 0;

green_norm2 = totalGreen2 ./ total_color2;
green_norm2 (isnan(green_norm2)) = 0;

red_mean2 = mean(red_norm2);
green_mean2 = mean(green_norm2);
red_std2 = std(red_norm2);
green_std2 = std(green_norm2);

read2 = double(imread('test.bmp'));

[Xs, Ys, Zs] = size(read2);
skin_detection2 = zeros(Xs, Ys);
for x = 1:Xs
    for y = 1:Ys
        red = read2(x, y, 1);
        green = read2(x, y, 2);
        blue = read2(x, y, 3);

        sum = red+green+blue;
        if sum > 0
            main_red = red / sum;
            main_green = green / sum;
        else
            main_red = 0;
            main_green = 0;
        end

        red_prob2 = gaussian_probability(red_mean2, red_std2, main_red);
        green_prob2 = gaussian_probability(green_mean2, green_std2, main_green);
        prob2 = red_prob2 * green_prob2;
        skin_detection2(x, y) = prob2;
    end
end
```

```
negative_histogram = read_double_image('negatives.bin');
positive_histogram = read_double_image('positives.bin');
frame20 = double(imread('test.bmp'));
result = detect_skin(frame20, positive_histogram,  negative_histogram);

msa = double.empty();
mnsa = double.empty();

msa2 = double.empty();
mnsa2 = double.empty();

msa3 = double.empty();
mnsa3 = double.empty();
for threshold = 0:0.1:100
    [x1, y1] = eval_module(skin_detection1, threshold);
    msa(end+1) = x1;
    mnsa(end+1) = y1;

    [x2, y2] = eval_module(skin_detection2, threshold);
    msa2(end+1) = x2;
    mnsa2(end+1) = y2;

    [x3, y3] = eval_module(result, threshold);
    msa3(end+1) = x3;
    mnsa3(end+1) = y3;
end
close all;
figure(1);
plot(msa, mnsa, 'k-');
hold on;
plot(msa2, mnsa2, 'r--');
plot(msa3, mnsa3, 'b:');
set(gca, 'XLim', [0 105]);
set(gca, 'YLim', [0 105]);
set(gca, 'XGrid', 'on');
set(gca, 'YGrid', 'on');
xticks([0:10:100])
yticks([0:10:100])
set(gca, 'PlotBoxAspectRatio', [1 1 1]);
legend('method1', 'method2', 'method3', 'Location', 'NorthWest');
```