Skin Detection with Pixel Chromaticity

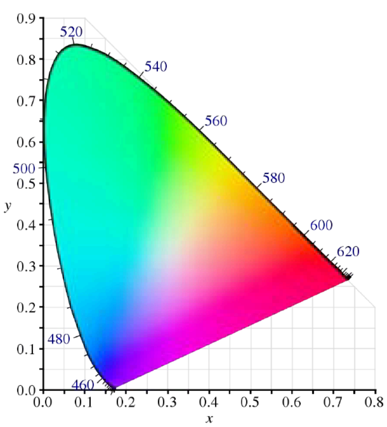
Team YOLO - s19 (205, 355, 408, 420)

1. Introduction

Skin detection is a crucial task in image processing and computer vision, with applications in face recognition, gesture analysis, and biometric authentication. This project aims to devise a simple skin color detection algorithm using chromaticity values extracted from images. By analyzing and modeling the color distribution of skin pixels, we can effectively differentiate skin from non-skin regions.

2. Key Definitions

Chromaticity



Chromaticity is a property that characterizes the color quality of an object seen by the human eye and gives a measure of how well an object is seen by the human eye. It is independent of the brightness and so is different from luminance where brightness is indicated. It can be determined by the spectral content of light.

The transformation from RGB to XYZ is given by,

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697} \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00 & 0.01 & 0.99 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

If we divide the XYZ values by the sum of X+Y+Z, we obtain the chromaticity coordinates,

$$x = \frac{X}{X+Y+Z}, \quad y = \frac{Y}{X+Y+Z}, \quad z = \frac{Z}{X+Y+Z},$$

Color Distribution

The statistical representation of colors present in an image is often modeled using mean and covariance.

Mean

The mean (also called the average) is a statistical measure representing the central value of a set of numbers. In image processing, the mean of a color distribution is the average color value across selected pixels. It is calculated as,

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

where x_i represents each pixel's chromaticity value, and N is the total number of selected pixels.

Covariance Measure

Covariance measures how much two variables change together. In image processing, it helps analyze how two color channels (e.g., red and green) vary relative to each other in a given region. The covariance between two variables X and Y is given by,

$$Cov(X, Y) = \frac{1}{N}\sum_{i=1}^{N} (X_i - \mu_X)(Y_i - \mu_Y)$$

where μ_X and μ_Y are the means of X and Y. A high positive covariance indicates that as one color channel increases, the other does too, while a negative covariance means they vary inversely.

Region of Interest (ROI)

A region of interest (ROI) is a portion of an image that you want to filter or operate on in some way. You can represent an ROI as a binary mask image. In the mask image, pixels that belong to the ROI are set to 1 and pixels outside the ROI are set to 0 .

3. Project Steps

I. Problem Identification & Project Planning
II. Data Collection
III. Algorithm Development
IV. Testing with a couple of images
V. Report writing

4. Methodology

The process of detecting skin using chromaticity values involves the following steps.

4.1 Image Acquisition

- Load the input image using OpenCV.
- Convert the image to a floating-point format for accurate chromaticity computation.

4.2 Chromaticity Computation

- Normalize the RGB values of the image.
- Compute chromaticity values:
    - $x = R / (R + G + B)$
    - $y = G / (R + G + B)$
- Ensure numerical stability by preventing division by zero.

4.3 Selecting the Skin Region of Interest (ROI)

- Manually select a skin area from the image using OpenCV's ROI selection tool.
- Extract chromaticity values from the selected region.

4.4 Computing Skin Statistics

- Calculate the mean chromaticity values (mean_x, mean_y) of the selected skin region.
- Compute the covariance matrix of chromaticity values.

4.5 Skin Region Detection

- Extract chromaticity values for the entire image.
- Compute the Mahalanobis distance for each pixel:
    - Calculate the difference between pixel chromaticity and mean skin chromaticity.
    - Compute the inverse of the covariance matrix.
    - Apply the Mahalanobis distance formula to classify skin pixels.
- Generate a binary mask indicating detected skin regions.

4.6 Post-Processing

- Apply morphological operations (opening and closing) to refine the detected skin mask.
- Use the mask to extract skin regions from the original image.

4.7 Visualization and Analysis

- Display the original image, binary skin mask, and detected skin regions.
- Generate a Kernel Density Estimation (KDE) plot to visualize chromaticity distributions.
- Evaluate the robustness of the method across different images.

5. Results and Discussion

The skin detection method relies on chromaticity, which helps minimize the impact of brightness changes but can still be affected by lighting conditions. The segmentation results seem effective, though some noise and missing regions indicate potential improvements in preprocessing (e.g., morphological filtering).

The chromaticity distribution supports the assumption that skin colors, despite differences in ethnicity, fall within a relatively small region in chromaticity space. The third image suggests that environmental factors influence chromaticity, which should be accounted for in robust skin detection models.

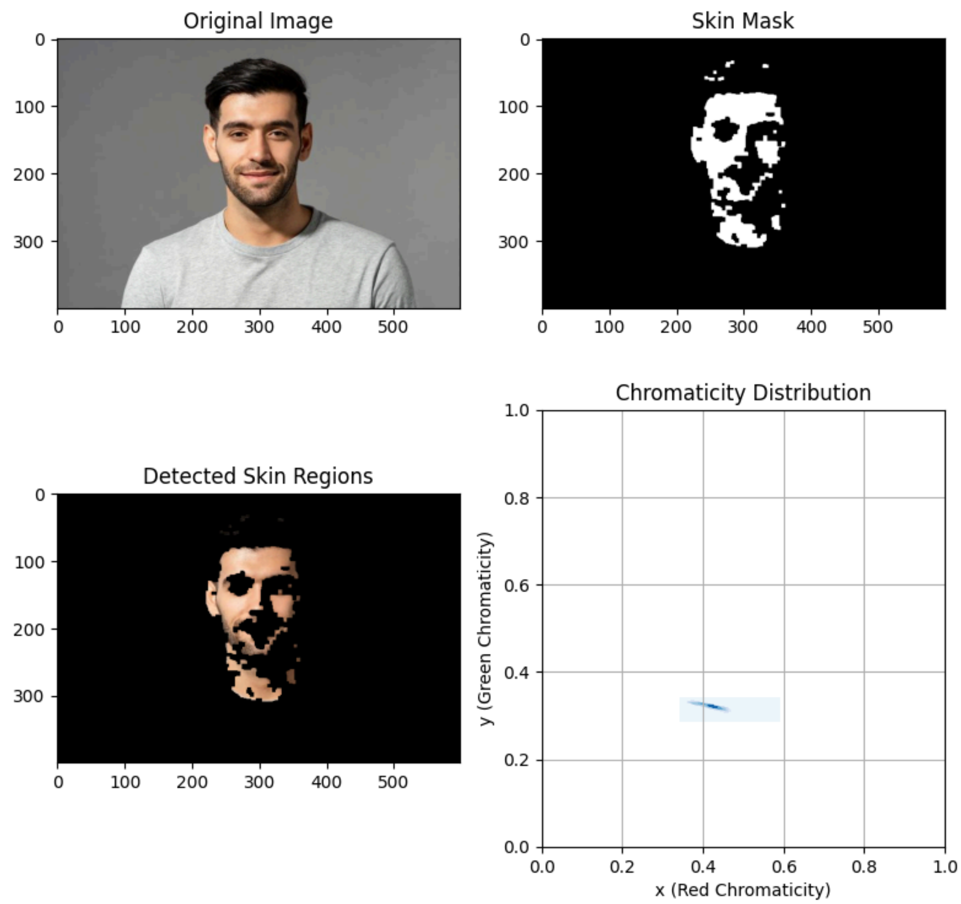


Figure 1: Detecting Skin By Using Chromacity Values with Mean and Covariance

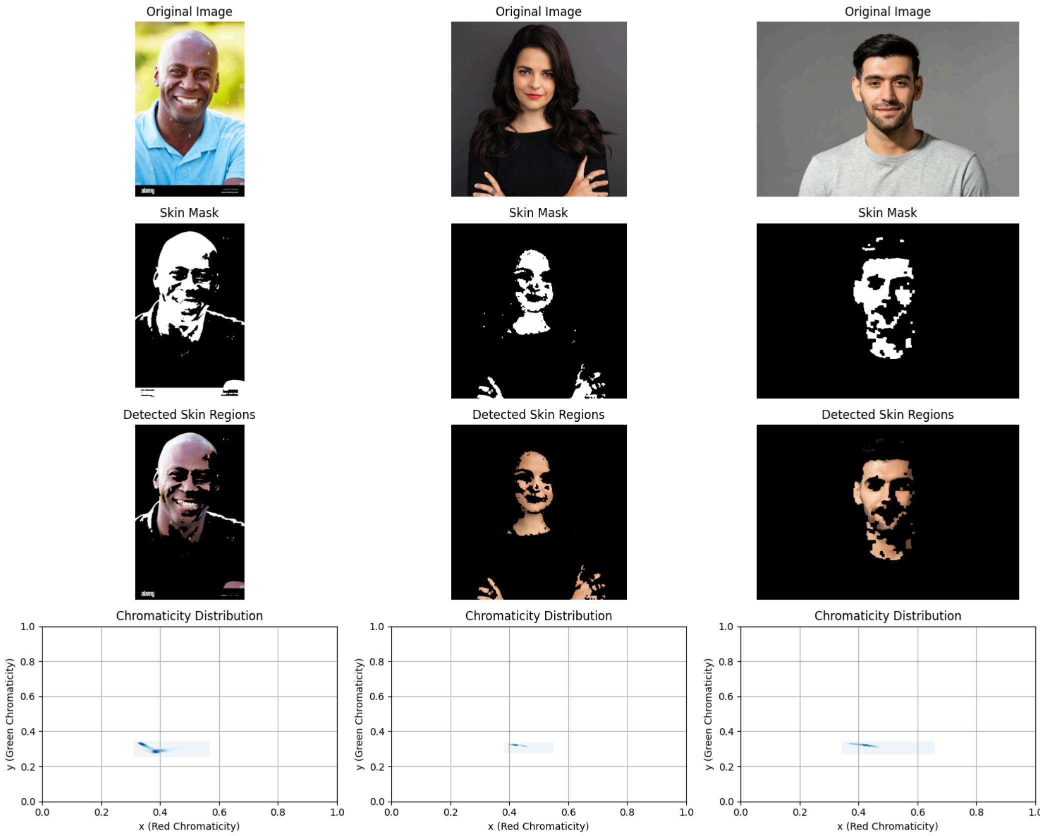


Figure 2: Testing for Several Images

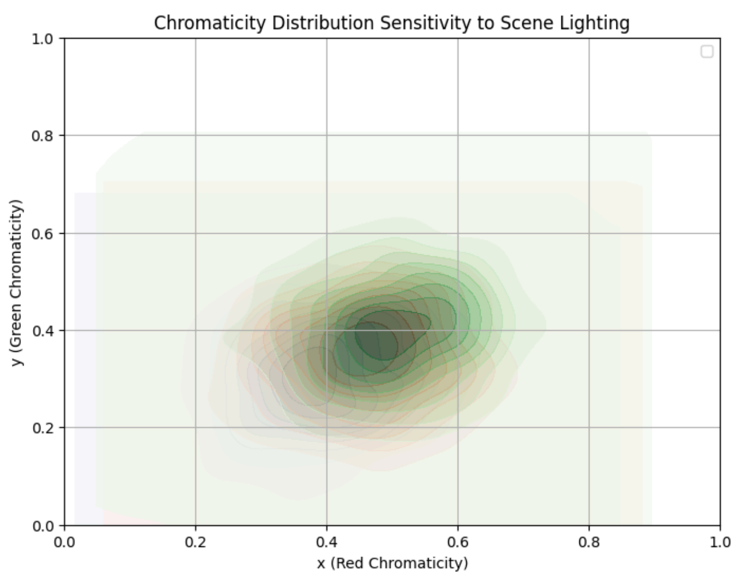


Figure 3: Final Evaluation for Figure 3

6. Task Breakdown

The project was divided into three main tasks,

Task	Team Member
Project Reporting & Documentation	S19205, S19420
Code Development, Continuation & Evaluation	S19355, S19408

7. Conclusion

The project successfully implemented a skin detection algorithm using chromaticity values and statistical modeling. The results demonstrated that chromaticity-based color modeling is effective in identifying skin regions. However, the algorithm's sensitivity to lighting variations highlights the need for adaptive color-balancing techniques. Future improvements could include more robust and generalized skin detection across diverse skin tones and environmental conditions.

8. References

1. Szeliski, Richard. *Computer Vision: Algorithms and Applications*.
2. MathWorks. "ROI-Based Processing." *MathWorks*, https://www.mathworks.com/help/images/roi-based-processing.html.
3. Ranzato, Marc'Aurelio. "Publication from CVPR 2010." *Computer Vision and Pattern Recognition (CVPR)*, 2010, https://www.cs.toronto.edu/~ranzato/publications/ranzato_cvpr2010.pdf.