

## **MP4 - Histogram-Based Skin Color Detection**

Logan Boswell

### **Overview**

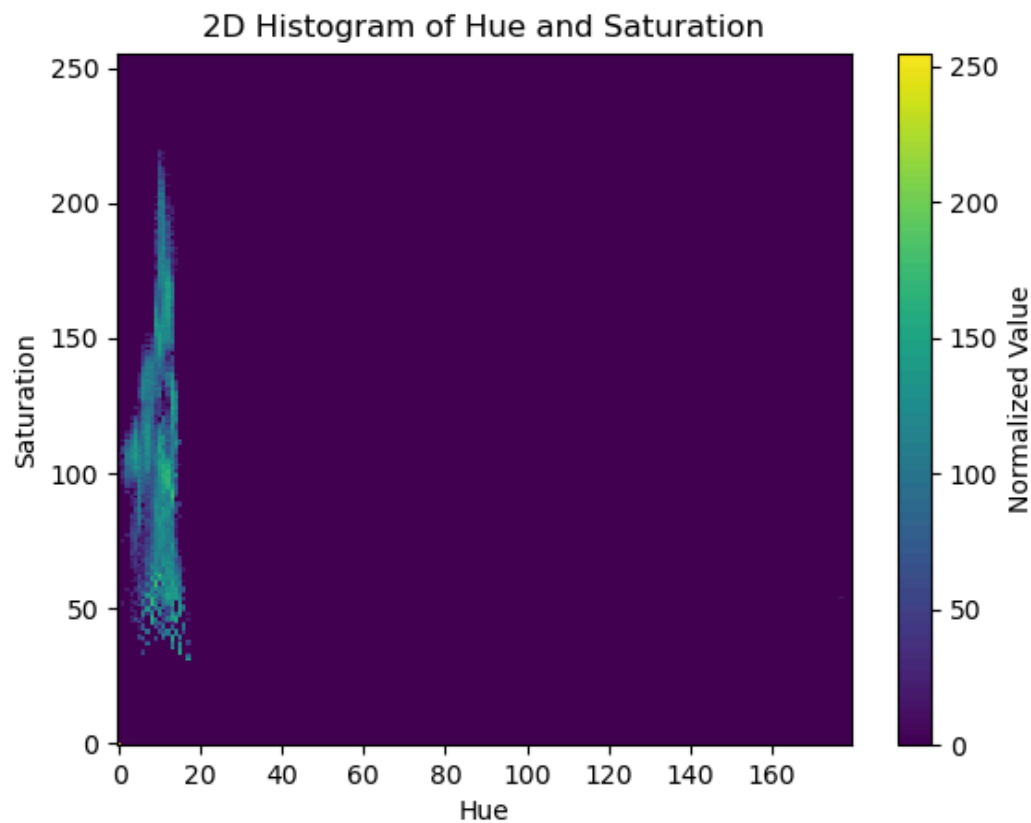
This MP is focused on implementing color-based segmentation to identify skin tones. In order to implement this technique, multiple python files were written to crop images for testing, generate a 2d histogram for skin tones data, and a function for detecting skin tones.

### **Cropping**

Before a 2d histogram could be generated to identify skin tones, some test images needed to be made. In order to do this, I wrote a python script that loops through the images in a certain folder and allows the user to crop multiple sections of an image and save them to use for training. The cropping process was made possible by using the RectangleSelector widget in the matplotlib.

### **Training**

The training process consisted of generating a 2d histogram based on pixel color values. In this case, I used the HSV color space. This script looped through all of the images in a training folder and added the H and S values to corresponding variables and used them to create a 2d histogram using `np.histogram2d()`. The 2d histogram generated is shown below. One important note to mention is that the histogram was converted to log scale since there was a large amount of data included in the histogram. After generating the histogram, it along with the two sets of bins were saved as a .npz file to be used when identifying skin tones.



### Skin Detection

For my `SkinDetection()` function, an image and .npz file are inputs. A blank image is defined and each pixel of the input image is checked compared to 2d histogram to see if the pixel is a skin tone. If the pixel is a skin tone, the associated pixel in the blank image is replaced by the pixel in the input image. After this process is complete, the blank image (which is no longer blank) is returned.

### Results

Here are the testing results with the given .bmp images.



