Ikhlas Attarwala / 10-17-2019 / MP #4
EECS 332 – Introduction to Computer Vision

This was a long but fun machine problem. For my results, you may run ***main.py*** via a python IDE, but I would recommend running ***main.ipynb***. When you do, you can test the functions ***final_test('gun1.bmp', 'rgb')*** and ***hist2dgraph('rgb')***.

- In both cases, 'rgb' is the code to run the RGB color space. Alternative color spaces that I used are 'nrgb' for normalized-RGB and 'hsi' for HSI.
- 'gun1.bmp' is an example file name.

This time we learned about skin detection, one of the most crucial topics of image analysis. It is an important and challenging problem in computer vision today. Often skin detection is used to be able to identify faces, as facial recognition is a very popular concept nowadays, but at times, body identification, such as when I was tracking football players for a school project, is also important and requires identification of limbs, joints, humanoid figures, and skin detection. Joy Buolamwini gave a talk very recently about algorithm bias, and one of the concerns she spread was the lack of training data in commonly used algorithms that detect differences in skin tone. Today we trained our program using a 2d-color histogram.

First, I, along with some classmates, took 40 pictures of 20 students (10 dark-toned and 10 light-toned individuals), and asked them to give different expressions, and tried to take pictures in different lighting environments. Next I manually cropped each image in the code so that I could get the unadulterated part between the eyes and bottom of the neck. Then I took each cropped image and converted it into a numpy array. This array gave me the RGB vaues of each pixel across the image, and I used this array to (1) normalize the RGB values, and (2) convert them into HSI values, as taught from class and our lecture slides (all formulas from lecture slides were used). Next I took each cropped image and separated them into dictionaries according to their respective color space, where the keys were the R-G values, and the values were the count of how many times that R-G value appeared in the training set. From literature, it appeared the blue value would not play a heavy roll in skin detection (same as intensity in HSI). I then transformed this into a 2d-histogram to observe my results. Fortunately, I got a beautiful result with my RGB color space (Figure 2). Using this 2d-histogram and my trained dictionaries, I set a threshold for the least likely values that should be representing skin. In the final step, I test every pixel from a testing image and see if it falls above the required threshold. If it does, it will stay on the image, otherwise it will be replaced with a black pixel (RGB 0,0,0). My results looked great, albeit a little noisy, but that's to be expected with only 40 pictures. If I had trained across hundreds or thousands of images, I'm sure I would have received better results. All 3 images were tested and I produced skin from each. You will notice on the following page that the sets of images look different (ex. HSI results look blue/green). I left this **on purpose** to showcase the importance of hue on an image, which isn't as easily transformative with RGB results. The 3rd color space I used was a normalized RGB vector space. Normalizing was an important step in the making of my 2d-histograms. The results show that I can detect skin with my program. I collected good data for the Gaussian-based color segmentation, but it ran for so long so I couldn't create Gaussians per color space, just had enough time for 1. I did not test with the Gaussian system. From my results of part 1, it appears the n-RGB has the most coverage, HSI covers the most while also having good detail and depth, and RGB last, but requiring less image manipulation. Images on the following pages!

.
.
.
.

Figure 0 – The 40 training images I used

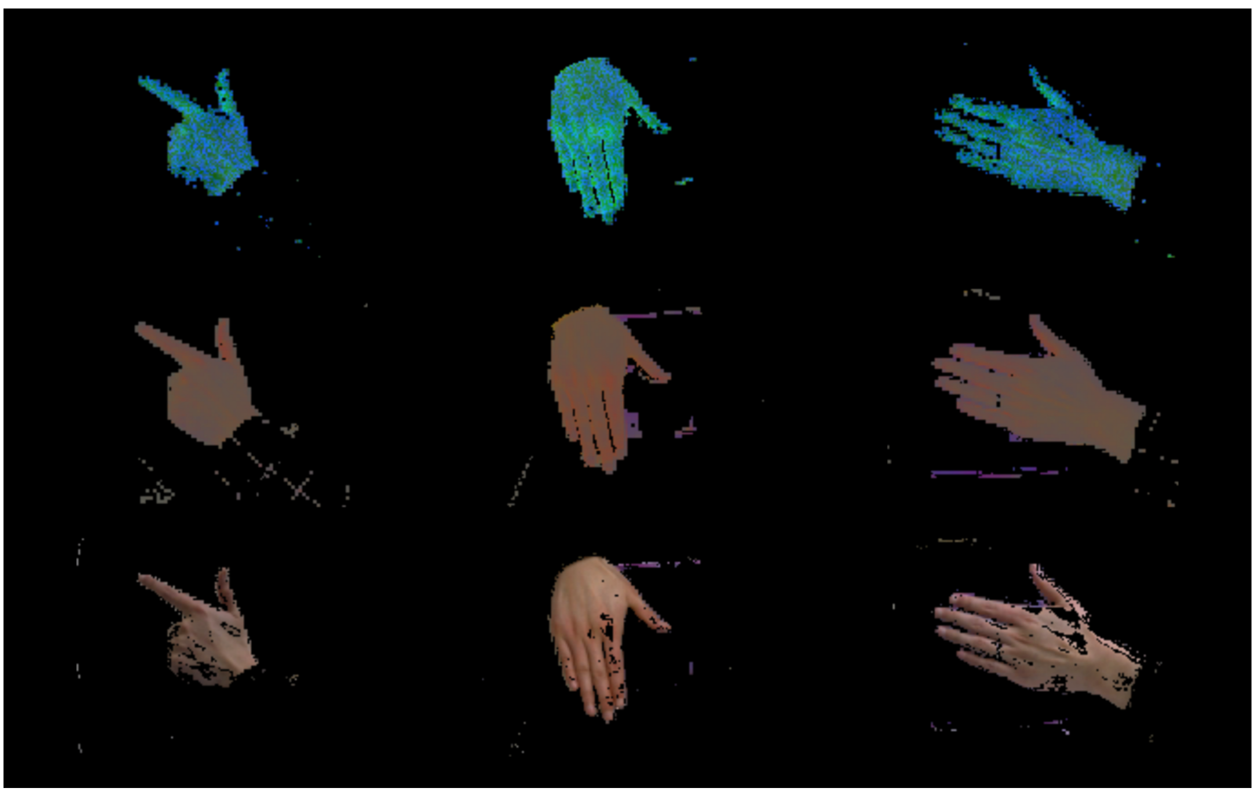

Figure 1 – The 3 color spaces (HSI, n-RGB, and RGB from the top) and results of my training for Part 1. Please note that I left the imbalanced hues/saturation on purpose to show the difference of the color space's activity.

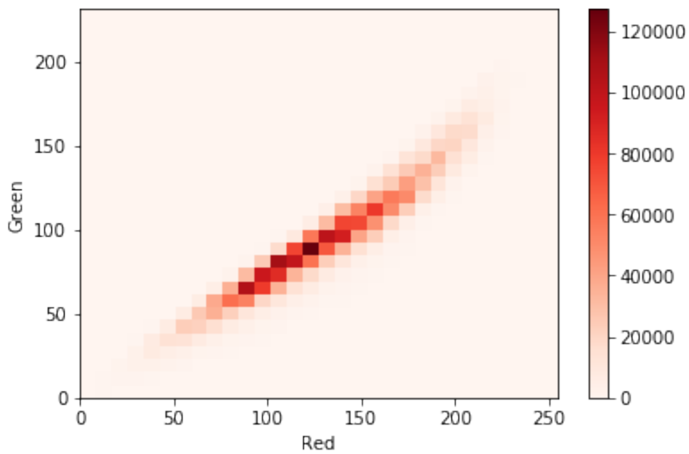

Figure 2 – RGB color space via 2d-histogram of training results. Beautiful!

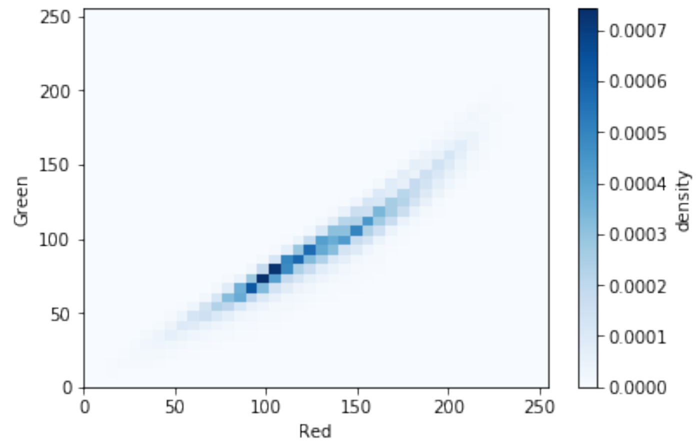

Figure 4 – RGB color space via Gaussian 2d-histogram training results

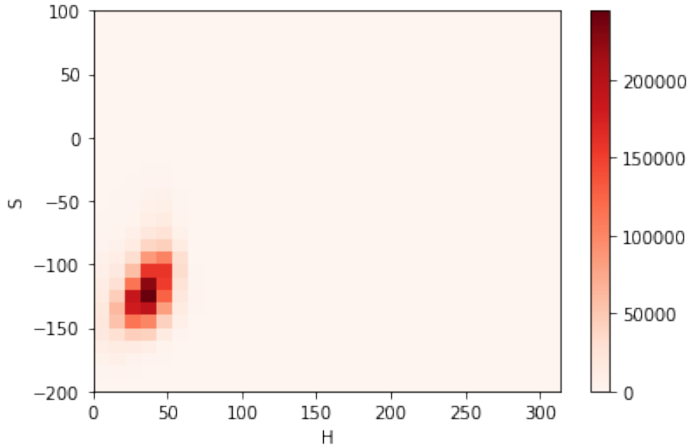

Figure 3 – HSI color space via 2d-histogram of training results

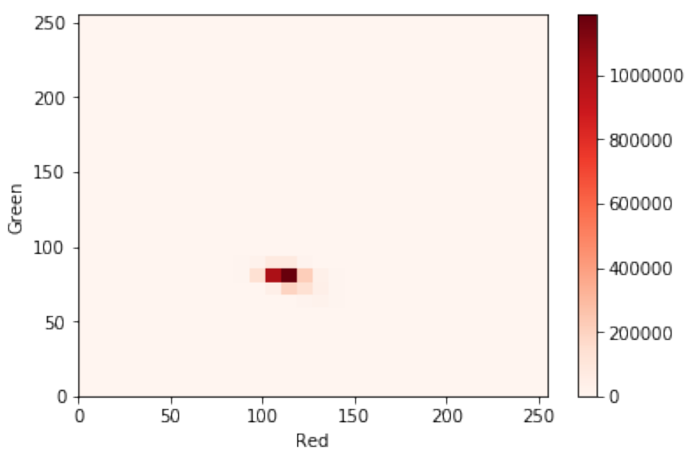

Figure 2 – n-RGB color space via 2d-histogram of training results