Machine Problem #3: Histogram-Based Skin Detection

In this lab, we were asked to train and implement a histogram-based skin tone detector. Such detectors are very popular due to their computational efficiency.

Training the detector (offline)

The Human Skin Image Database provides a wide selection of high-quality skin color images designed explicitly to be used as training data for projects like this.

To construct the training histogram, I iterated through these images, constructing two Python dictionaries for histograms of the RGB and HSV color spaces respectively, each with the general form

```
hist_rgb = { (r, g, b): pixel_count_1, ... }
```

or

```
hist_hsv = { (h, s, v): pixel_count_2, ... }
```
.

`mp4_histogram_training.py` performs this recursively on any directory of images provided to it, and then exports the dictionaries to `histogram_data/hist_hsv.pickle` and `histogram_data/hist_rgb.pickle`. (`import pickle` is a Python library for saving Python objects to binary files that can then be loaded in as-is for use in other Python programs.)

2D histograms are sufficient for this project, so I then took those dictionaries and iterated through their keys to create simpler dictionaries with only 2 of the 3 elements needed in them. Hence how we get

```
hist_rg = { (r, g): pixel_count_1a, ... },
hist_rb = { (r, b): pixel_count 1b, ... }
```

and so on. These 6 dictionaries are what were actually used for the later histogram creation.

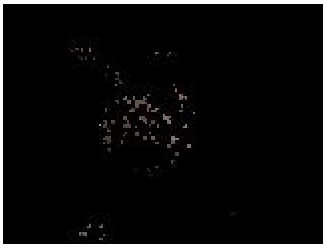
mp4_histogram_training.py can take a long time to run. It did however provide very good results; even with an absolute minimum threshold more pixels than not were correctly sorted.

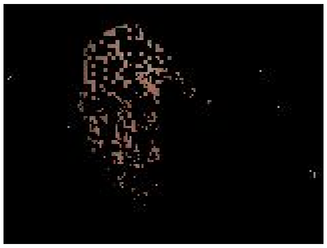
Selecting a good color space

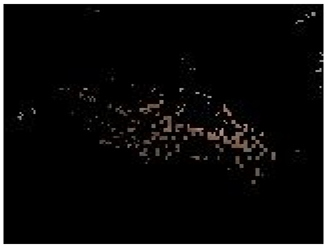
Experimenting with hist_rgb and hist_hsv, I found that I preferred working with HSV to RGB. Using a minimum threshold, HSV was able to detect more skin color regions than RGB, so I decided to stick with that one.

After that, I experimented between which two dimensions of hist_hsv did the best job. Out of hist_hs, hist_hv, and hist_sv, I found hist_hs to perform the best. The value of skin color can change a lot depending on the ambient brightness, but the hue and saturation seem to be more constant.

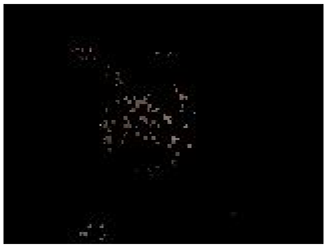
Color space: HSV, min threshold

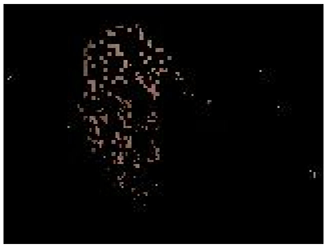


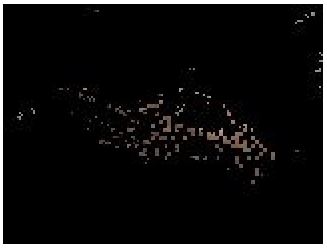




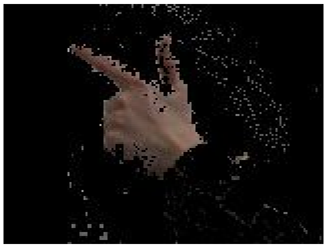
Color space: RBG, min threshold

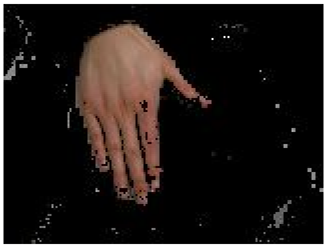


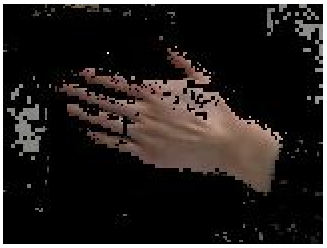




Color space: HS, min threshold



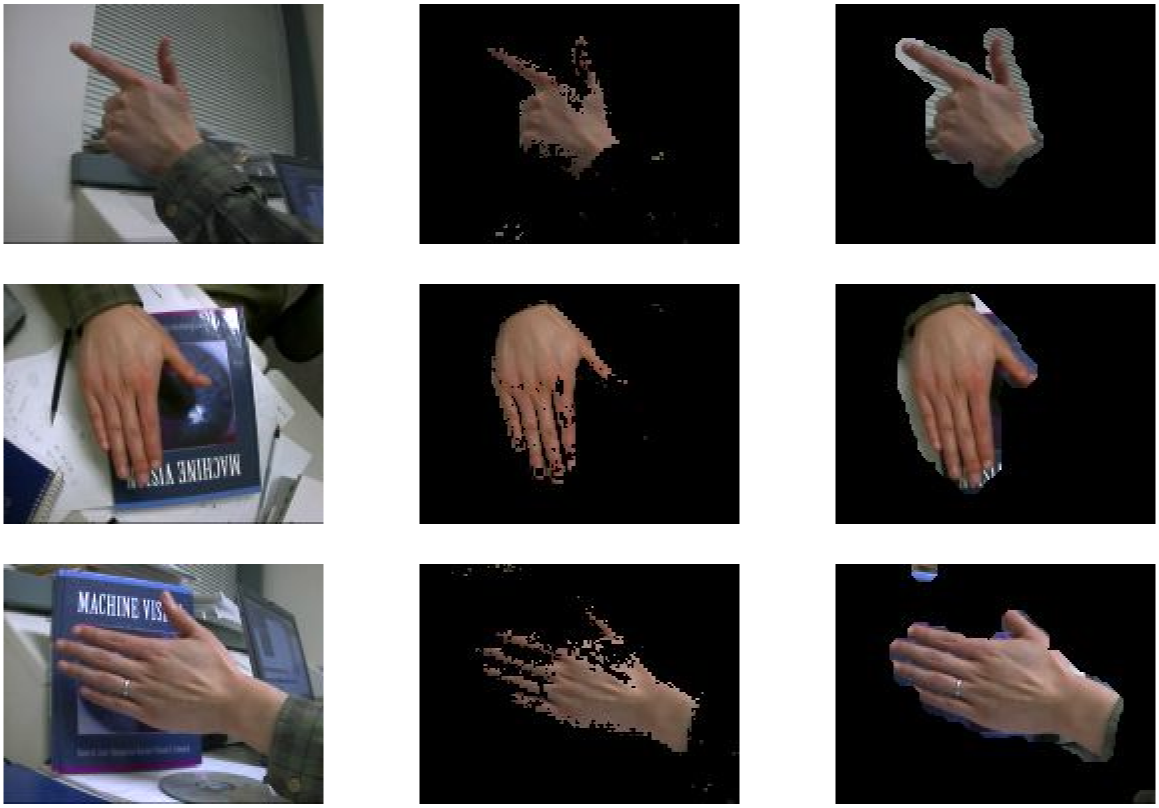




Finding skin regions in test images

My histogram collection was very pure, so my threshold was set very low at 0.00005 prevalence in the histogram or higher. The results are shown below.

I also used some of the erosion and dilation code from mp2.py to clean up the image and give it some cleaner boundaries. I considered adapting some of the code from mp1 to only pass through the largest contiguous region, but I was running out of time so I left it as is.



Original image **Post-histogram** **Post-morphological ops**

A full archive of the code, as well as the results from running my algorithm, have been uploaded alongside this report.