Stephen Ferro

MSAI 495

4/18/2024

Machine Problem 4

Introduction:

The goal of this assignment was to create and train a K-Means histogram model to detect and segment skin tones in images. Images of a hand were the inputs to the model and the output was binary images where skin tone pixels were replaced with white and all other pixels were replaced with black.

Algorithm:

Skin Region Selection and Pixel Collection:

1. For each training image, select regions of skin tone pixels.
2. Collect pixels from these regions and aggregate them into a list of skin pixels.
3. Train a k-means model using the collected skin pixels to find clusters representing different skin tones.
    a. Randomly assign cluster centroids initially.
    b. Calculate the Euclidean distance between each skin pixel and all cluster centroids to determine the nearest cluster for each pixel.
    c. Assign each skin pixel to the cluster whose centroid is closest, grouping pixels that are similar in color.
    d. Recalculate the position of each cluster's centroid by averaging the colors of the pixels assigned to that cluster.
    e. Repeat until the model converges.

Skin Detection in Test Images:

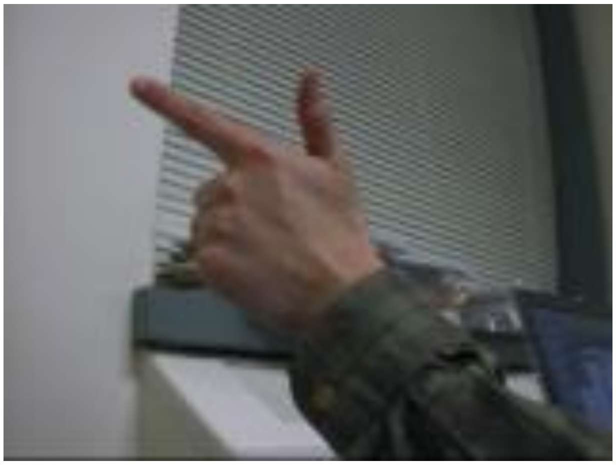
1. Apply the trained k-means model to each pixel in the photo to determine if the pixel matches one of the skin tone clusters.
2. Create a binary mask where white pixels indicate that the pixel in the photo matches one of the clusters in the k-means model.
3. Apply morphological operations (closing and opening) to clean up the skin mask.

Results:

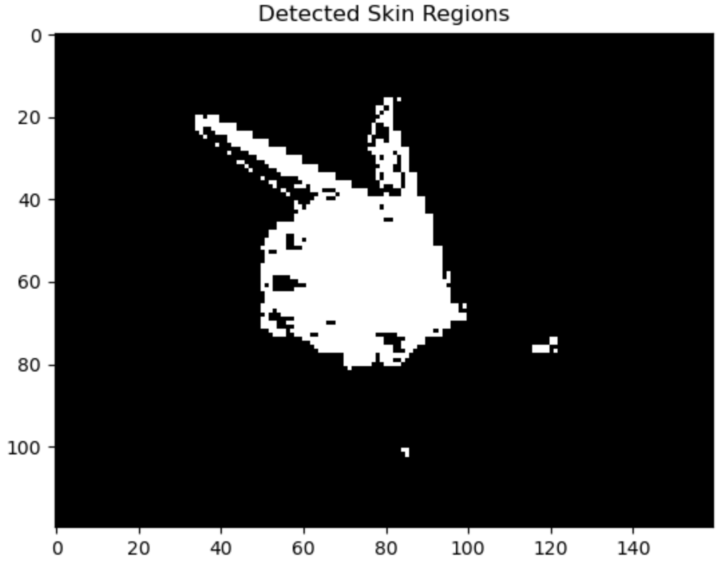
Overall, I was able to extract good results from all the photos using my model. I selected several skin-tone regions on each hand during the training phase to get the best results. I found that using the HSV color space resulted in the best segmentation, while using the RGB colorspace gave mixed results, especially in the gun.bmp image. For training the k-means model, I found that using 10 clusters worked well while still allowing the program to execute and train qucikly.

Though the segmentation was not perfect, applying opening and closing to the image output very clean masks for all images. For this reason, I consider the segmentation successful.

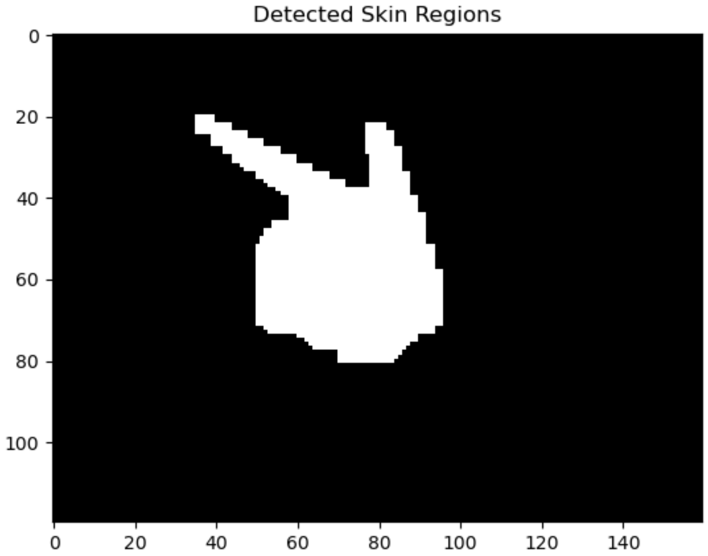
Gun.bmp:



Input Image

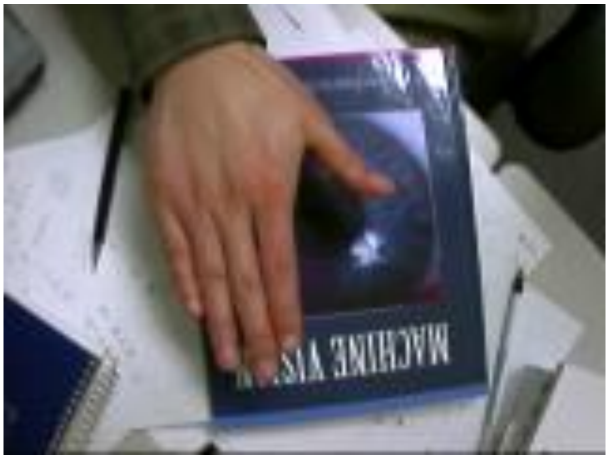


Model Output

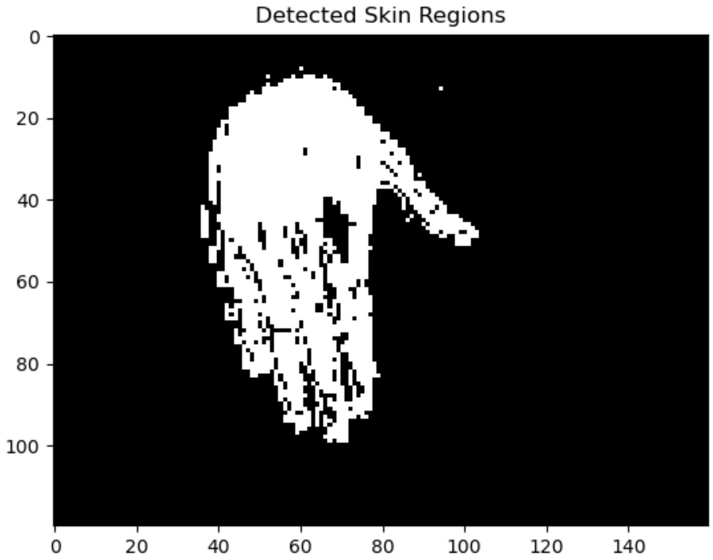


Output with Opening and Closing Applied

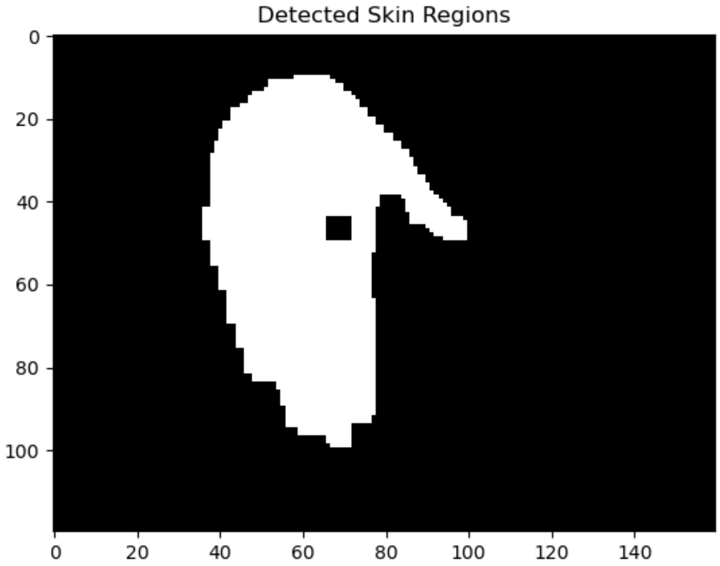
Joy.bmp:



Input Image

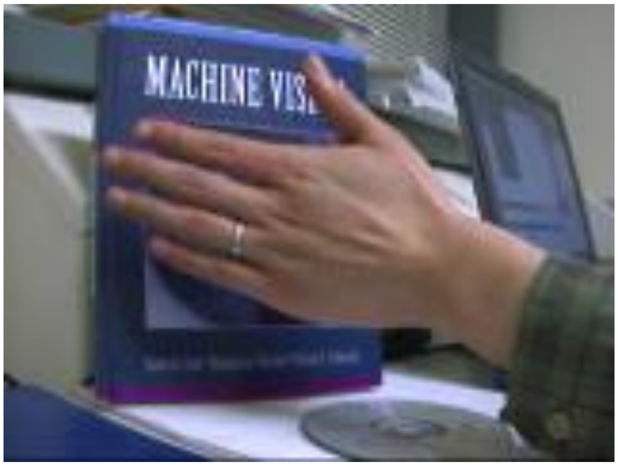


Model Output

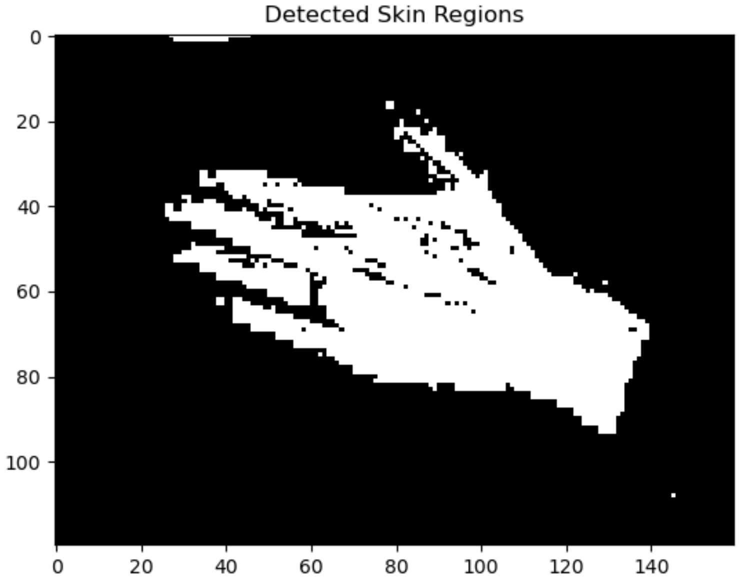


Output with Opening and Closing Applied

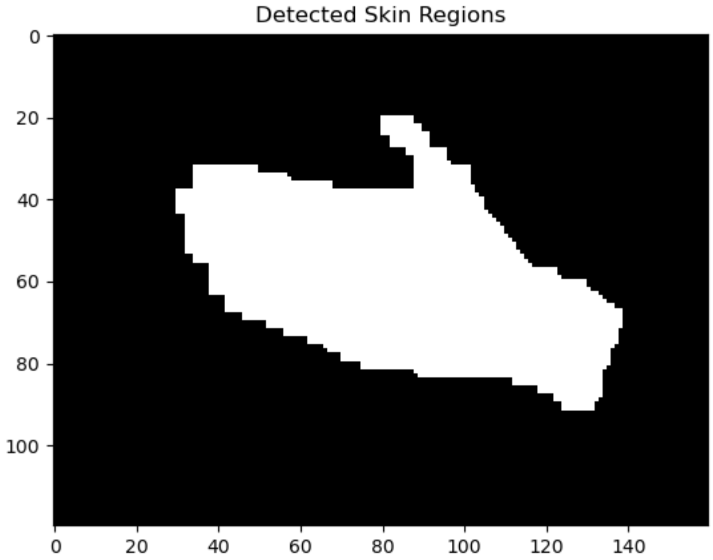
Pointer.bmp:



Input Image



Model Output



Output with Opening and Closing Applied