CENG466, Fall 2022, THE 3

1st Fırat Ağış
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
e2236867@ceng.metu.edu.tr

2nd Robin Koç
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
e2468718@ceng.metu.edu.tr

***Abstract*—**

***Index Terms*—**

I. Face Detection

For face detection, we propose an algorithm that use k-means to separate color groups, chooses the color group that is most likely to include skin colors and then uses morphological operations and convolution operation to detect areas that are most likely to include faces in the image. Our algorithm posses 3 distinct stages:

1) *Preprocessing:* Process the image to be more suitable for our algorithm and performs some processes to make the algorithm run a lot faster.
2) *K-Means:* Using k-means algorithm with randomized starting means to detect the pixels that are most likely to be in the color of skin tones.
3) *Context Based Postprocessing:* Using morphological operations and convolution operation to detect which clusters of the pixels that are found in the previous step to be the part of a face.

A. Preprocessing

1) We first read the relevant image.
2) (Optional) We perform histogram equalization to differentiate skin tones from similar colors in the image, most notably, the leaves in the 1_source.png and shirt and hair in 3_sources.png.
3) (Optional) We down-sampled the 2_source.png by a factor of 9 in order to complete the algorithm in a reasonable time.
4) We bit-sliced all images, eliminating the least significant bit. Colors that are differentiated by a single bit are most likely to be in the same color group, meaning this operation affected the quality of the k-means algorithm minimally while reduicing the size of the color space by a factor of $2^3 = 8$.

B. K-Means

C. Context Based Postprocessing

1) (Optional) While processing 1_source.png and 3_sources.png, we used the assumption of faces being the central focus of images to eliminate pixels on the outer parts of the image.
2) We used opening operation with a 5x5 element to eliminate noise from the image.
3) We used closing operation with a 5x5 element to fix regions that are broken up during the previous step.
4) We used convolution with a elliptical mask, mimicking the shape of he face to determine locations where a face may lie.
5) We used thresholding to the result of the convolution, including the parts of the image that passed the tresholding, concluding the algorithm.

D. Results

II. Pseudo-coloring

III. Dependencies

We used following libraries for the described reasons.

- **os:** Handling non-existent input or output paths.
- **math:** Performing square root operation.
- **numpy:** Executing array and matrix operations.
- **PIL:** Reading images and converting them to arrays.
- **mathplotlib:** Creating histograms as graphics and writing arrays as image files.