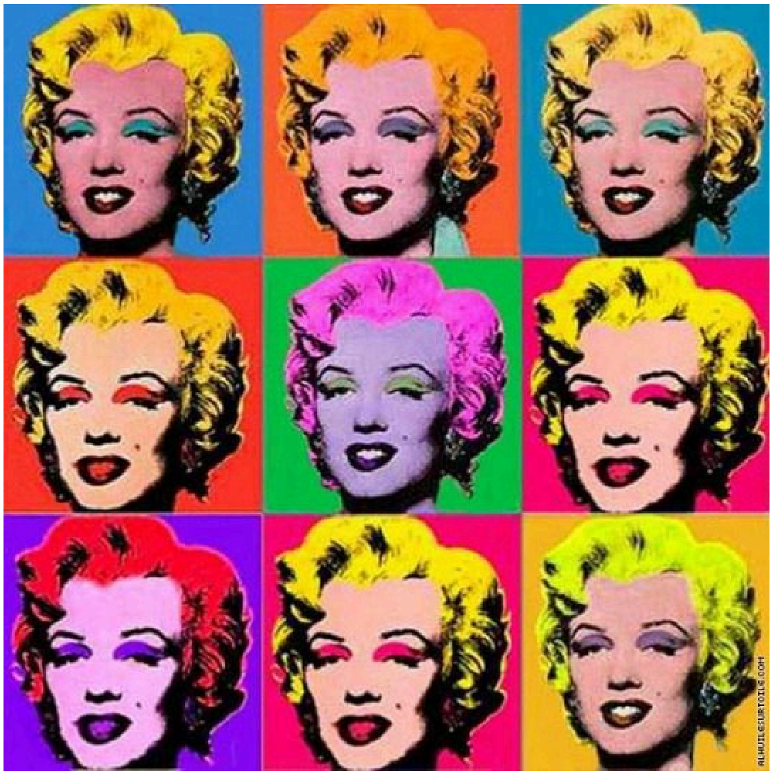
CS 4475 Project 3

Stuthi Bhat: sbhat307@gatech.edu
Victoria Choi: vchoi3@gatech.edu
Steven Li: sli862@gatech.edu
Raymond Liu: rliu443@gatech.edu

High Level Overview

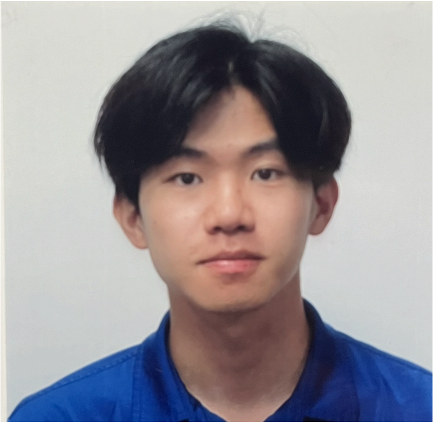
Our project takes in a photorealistic image of a human face, and turns it into a creative rendition of Andy Warhol's famous "Marilyn Monroe" painting. We found Warhol's messages about often forgotten humanity in a world of materialism, commodity, and celebrity interesting; that even behind the most superficial concepts lies deeper meaning and significance. We wanted to present our own take on a similar concept. We decided to render our own photos into the likeness of his art piece, but with more intense colors to symbolize that we are all more complex and multidimensional than is visible at a simple glance.

Given an input image, the code first uses haar-cascade face detection to crop the image to the largest face in the input image. Then, the image is mapped to a color space containing around 64 colors. Morphological transformations are then used to reduce noise and connect blocks of color, specifically 3 iterations of erosion followed by an iteration of dilation with a kernel size of 5. Finally, the blocks of colors are altered to produce the characteristic popping style of Andy Warhol's painting. To do this, each block of color is stored in a dictionary and mapped to another slightly altered, brighter color. To attain the color it is mapped to, each value in the BGR channel is shifted by a random value before converting to the HSV color space and scaling up the saturation value, preserving the core color but brightening the pixels. This process is performed 9 times with different colors before finally being collaged into a 3x3 grid. This process produces a final collage resembling Warhol's painting, but with even more intense color rendering.

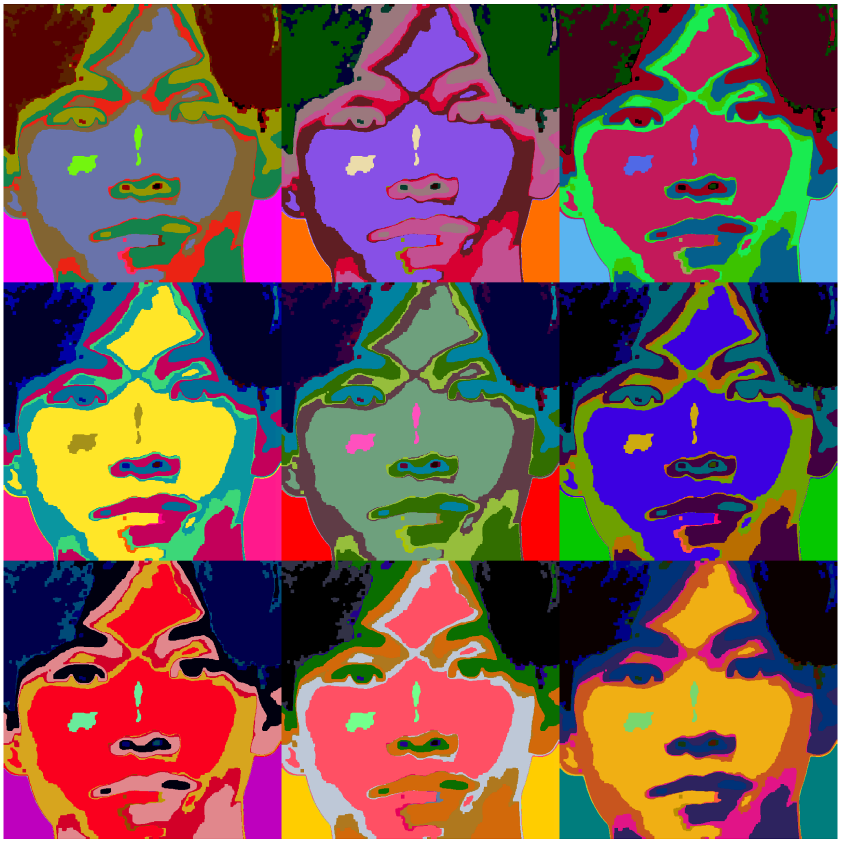


"Marilyn Monroe". Andy Warhol, 1962
Image attributed to The Andy Warhol Museum

Input Images



Output Images



Low Level Description

This section outlines the low level workflow procedure of our project. It will be delineated in steps.

Inputs: An image containing a clear face that you would like to work with. If there are multiple faces in the image, the largest one in the frame will be selected.

1. Enter the image of interest into the indicated location in the python script.
2. Enter the BGR colors you would like each of your images in the collage to be based around into colorsList.
3. The following steps are repeated 9 times on the input image and stored in a list, each one using a different color from the colors list:
    a. First, the cropFace function is applied to the image. This function identifies the largest face in the frame using Haar-cascade trained data and returns a cropped image containing only the face.
    b. Next, the mapToBlockColors function is applied to the image. This function maps the function to a color space of only about 64 colors, creating larger blocks of colors. Morphological transformations are then used to reduce noise and connect blocks of color, specifically 3 iterations of erosion followed by an iteration of dilation with a kernel size of 5.
    c. Next, the mapColors function is applied to the image to produce the characteristic popping style of Andy Warhol's painting. To do this, each block of color is stored in a dictionary and mapped to another slightly altered, brighter color. To attain the color it is mapped to, each value in the BGR channel is shifted by a random value before converting to the HSV color space and scaling up the saturation value, preserving the core color but brightening the pixels.
4. Finally, the collage function is used to put all images in the list into a 3x3 collage. This uses indexing techniques to copy over blocks into an array 9 times the size of the original.

Output: A custom artistic image made of a 3x3 grid of edited images, resembling Andy Warhol's "Marilyn Monroe."

Referenced Materials

We used the Haar-cascades data set for facial recognition, which is a trained dataset that is used to identify faces. We also simply used OpenCV documentation to understand some of the new functions we were using.

Haar-Cascades
https://github.com/opencv/opencv/tree/master/data/haarcascades

Morphological Transformation
https://docs.opencv.org/4.x/d9/d61/tutorial_py_morphological_ops.html

Changing Color Spaces from BGR to HSV
https://docs.opencv.org/4.x/df/d9d/tutorial_py_colorspaces.html

Point Breakdown

We will accept a 50/10 artistic to technical breakdown.