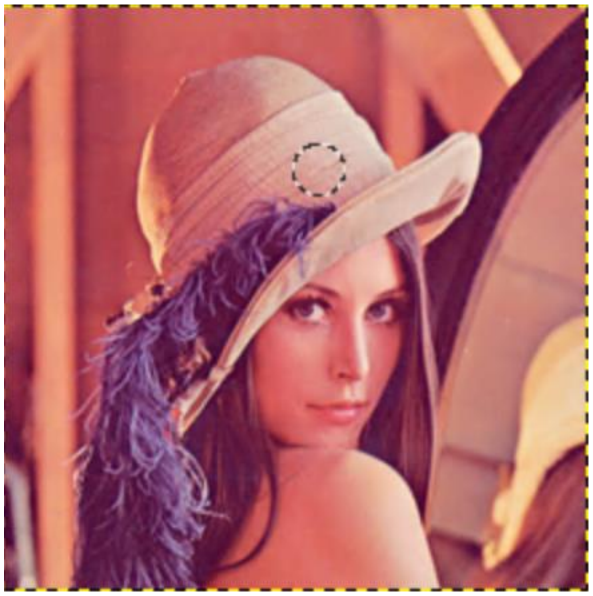
Brady Messer (wmesser)
CPSC 4040 Fall 2019
Final Project

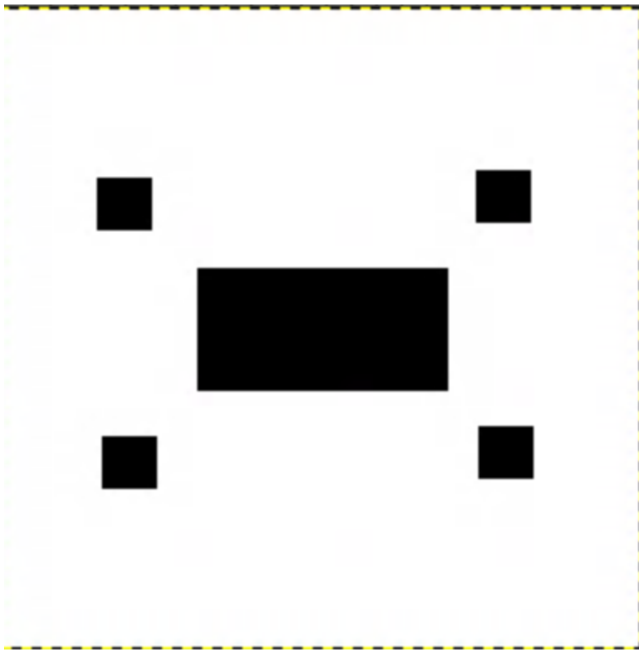
Color palettized images + image manipulation

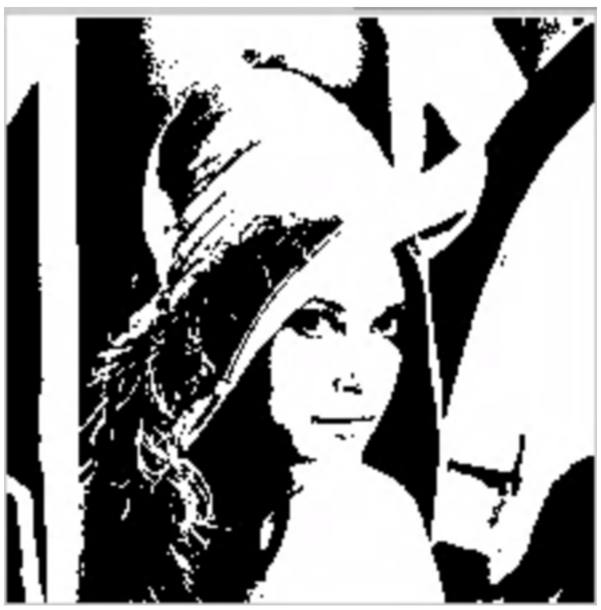
Description:

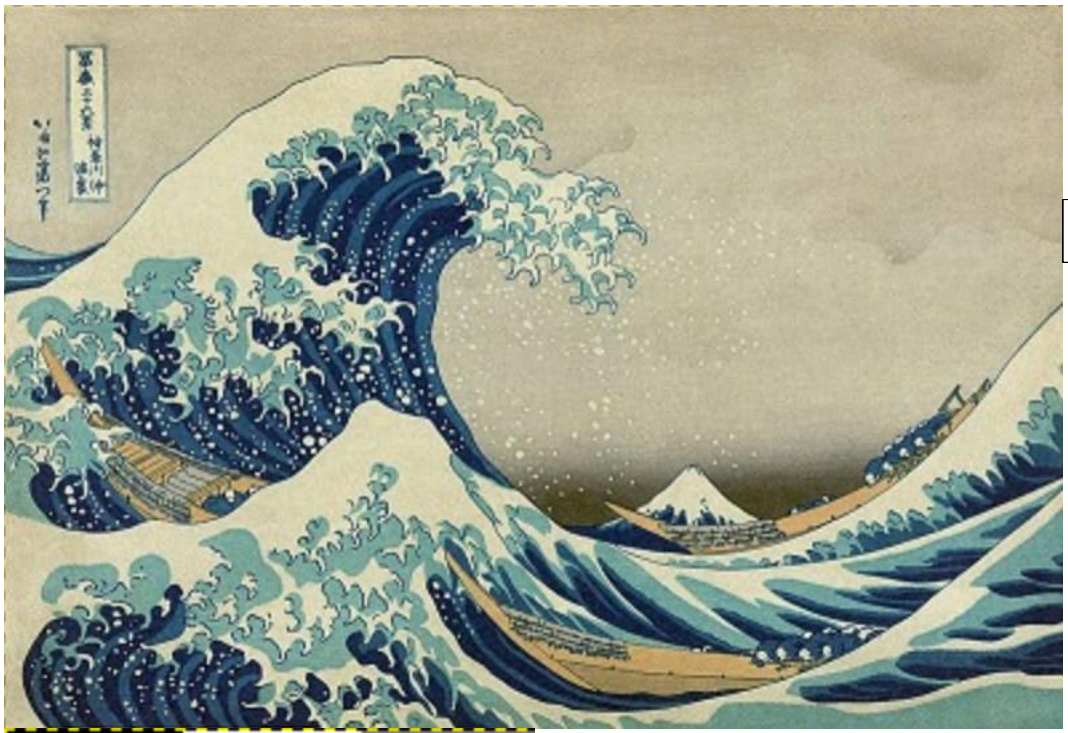
- This program takes in a limited color palette and applies it to a specified image. A color palette can be generated in one of two ways, by either manually creating one and saving it in a file, or by taking all the unique pixels of an image and using these as the color palette. The program works by comparing the pixels of the input image to the pixels of the color palette and replaces the pixel of the input image with the closest pixel to that color from the palette. The image manipulation functionality is a result of previous projects we completed in this course.

Examples:

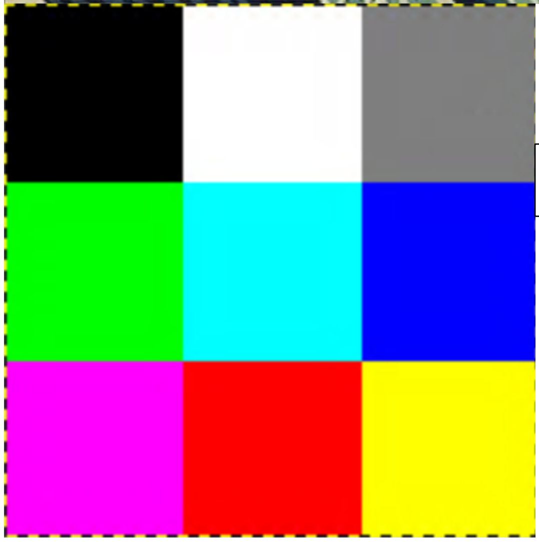

Input image


Color palette image

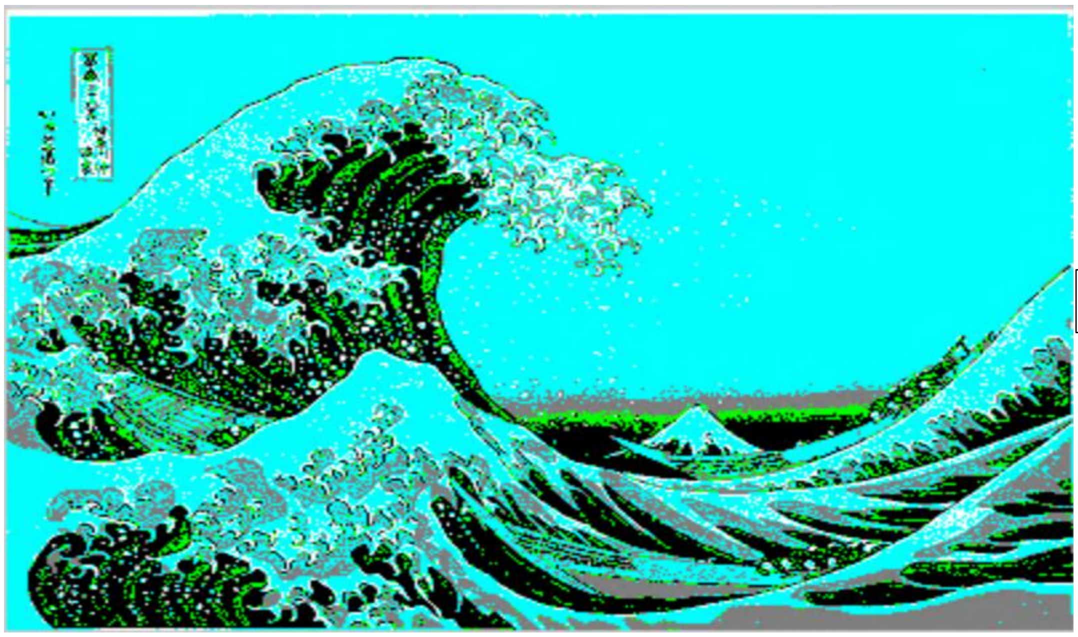

Resulting image



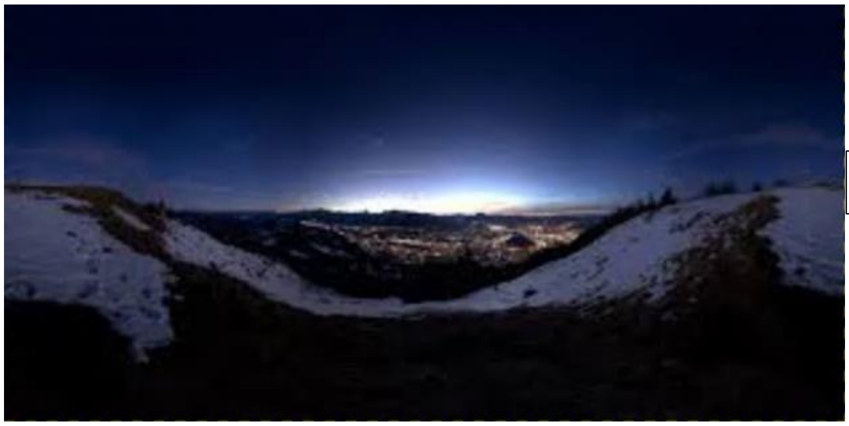
Input image

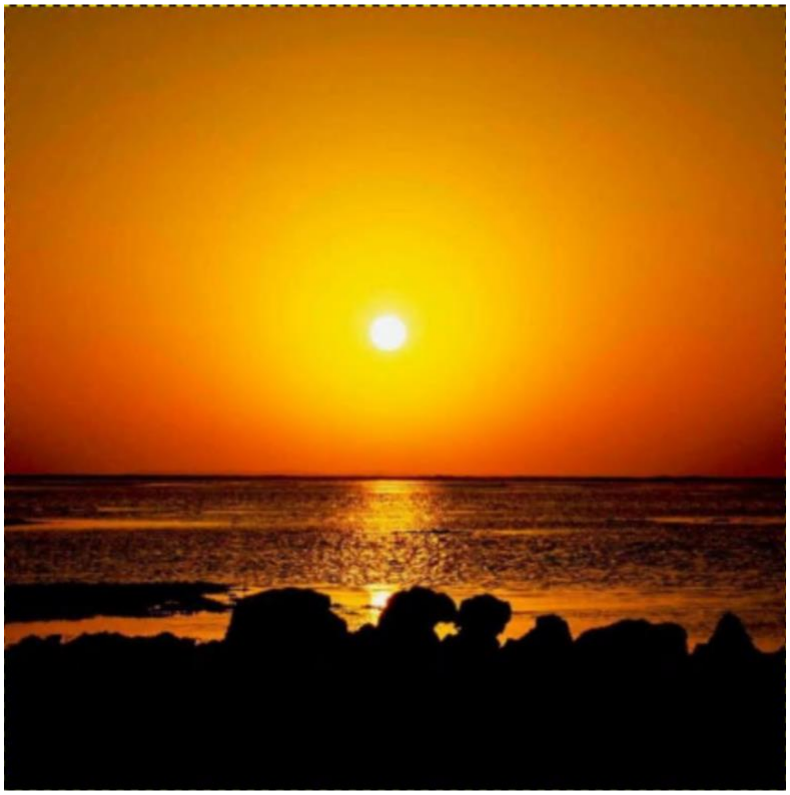


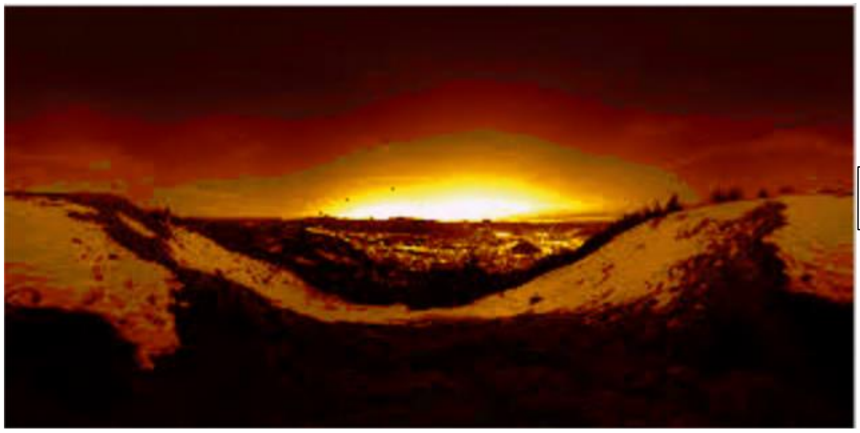
Color palette image

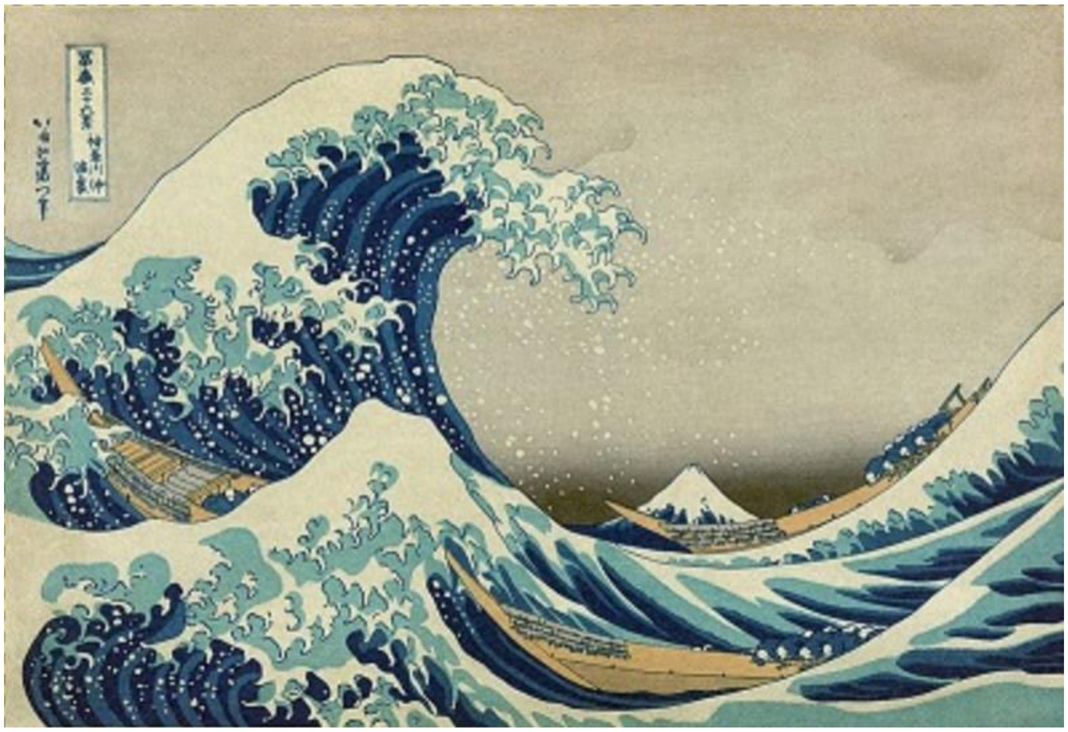


Resulting image

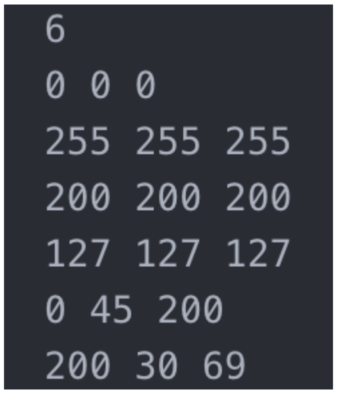

Input image


Color palette image


Resulting image

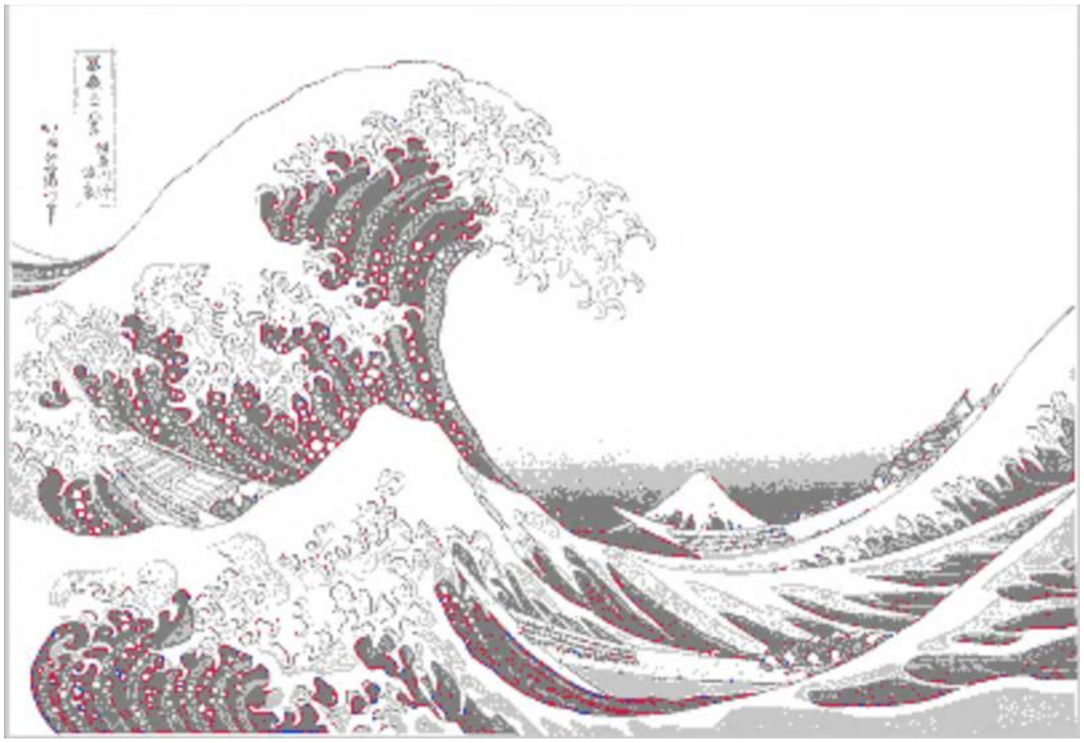


Input image



```
6
0 0 0
255 255 255
200 200 200
127 127 127
0 45 200
200 30 69
```

Manually created color palette



Resulting image

In addition to the color mapping functionality, the program also serves as a basic image manipulation program on top of that. Some of the other functions you can do after palettizing an image are convolution, shearing, rotating, scaling, perspective warping, translating, and flipping an image. A user could also invert the colors of the image and save any manipulated image to an output file. If you would like to explore any of these additional features, see the README.txt for instructions.

The approach I took can be outlined with the following algorithm:

1. Take in a source image
2. Generate a collection of acceptable rgb pixel values via any method you wish (I chose two methods, manual specification or generation from another image)
3. Using this collection of pixels, iterate through the source image and for each pixel find the closest matching color from the collection of acceptable pixels, and replace the source pixel with the closest matching palette pixel
    a. To find the closest matching pixel I took a simple approach and subtracted the r, g, and b values from the source image from their corresponding counterparts in the acceptable pixel collection. Then I summed these three values. The pixel with the lowest absolute value difference was deemed to be the closest in color.