Cartoonify

By Claire Bulick

Project Description

"Develop an application that makes cartoon-like images from pictures."

Drawn by Marco D'Alfonso for FOX Sports

Project Description

Requirements:

1.) Upload a photo and convert it to a cartoon-like image.
2.) Examine various algorithms and consider designing your own.
3.) Control the contrast, tint, and intensity of the cartoon.
4.) Allow the user to specify the number of colors in the cartoon image.
5.) Allow the user to specify which colors will be in the cartoon image.
6.) Implement save and open functions.

Solutions

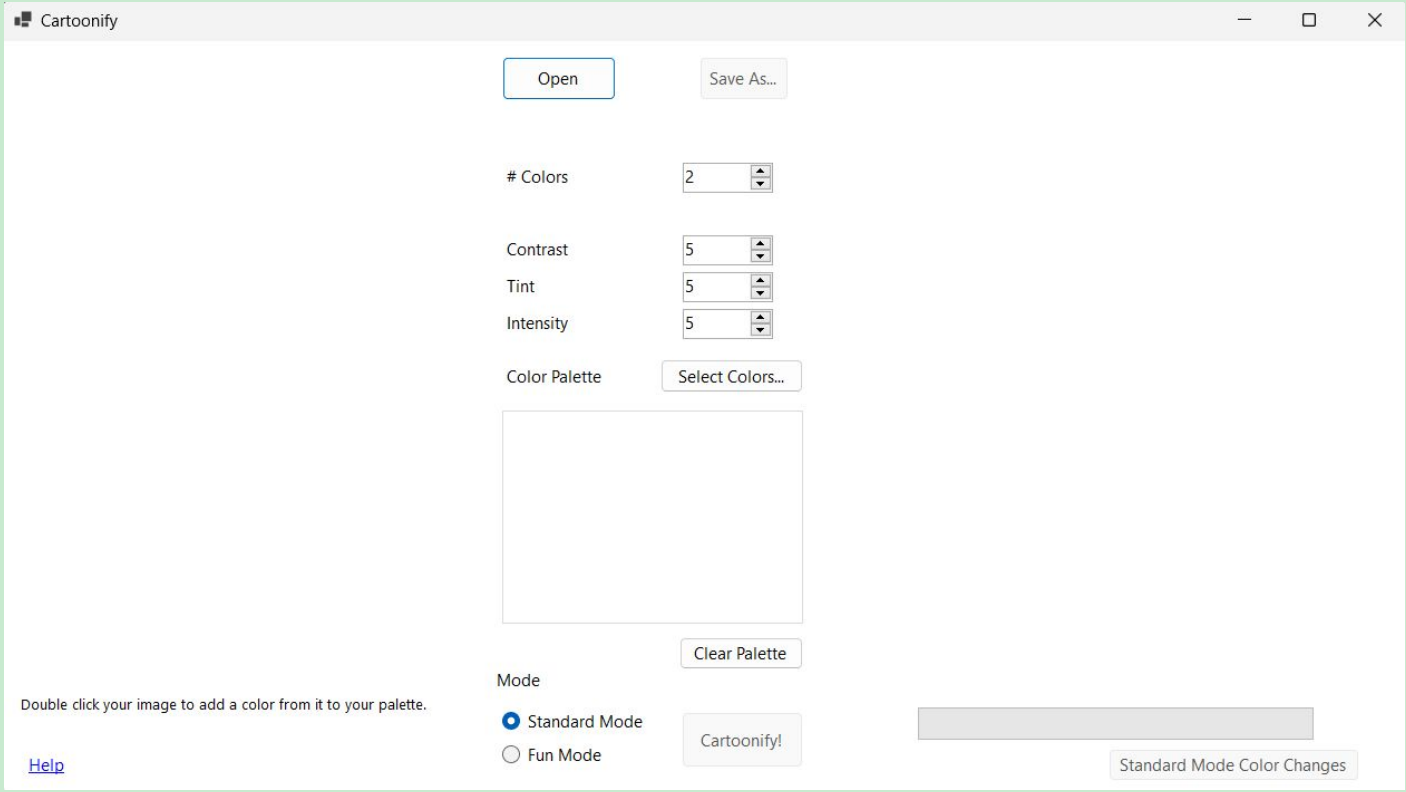
Width: 474px
Height: 315px

-C# Windows form application
-Pixel-by-pixel image processing
-Finding most similar colors
-Many arrays, Color objects, and Lists of objects

Solutions

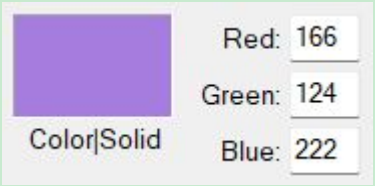
-Providing areas for users to add image, adjust settings, etc.

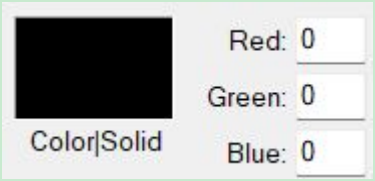
-The coloring algorithm takes values from these areas



How do colors work?

Color objects: A, R, G, B parameters, all 0-255

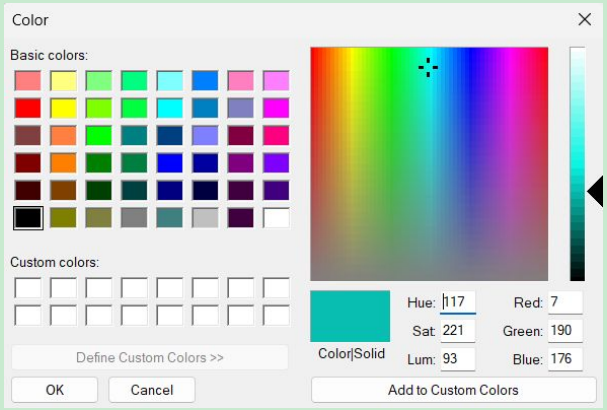




RGB is not the only way this program looks at color...

-There are two ways that colors can be added to the palette:

Eyedropper/Color Sampler

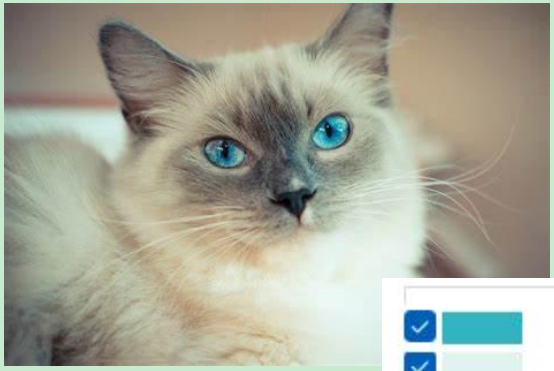


Custom Color Picker

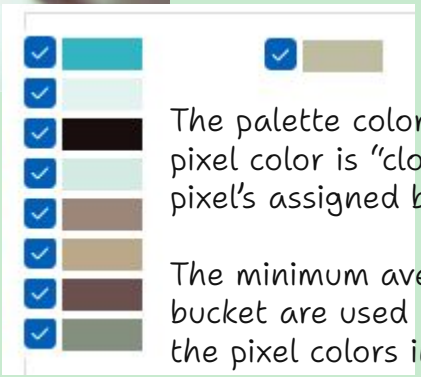
Colors are added to a List object, which then appears as the palette

Grouping Algorithm: K-Means

- Makes clusters ("buckets") based on the lowest averages
- Uses those averages on the next pass to update the clusters



Cycle through palette colors and find the minimum average...



The palette color that the current pixel color is "closest" to becomes that pixel's assigned bucket.

The minimum averages for each bucket are used for comparisons with the pixel colors in the next pass.

7	7	7	8	7	7	7	7
7	8	4	8	8	8	8	7
5	4	3	4	8	8	8	8
5	4	3	4	8	8	8	8
5	4	1	3	4	8	8	8
2	4	3	1	4	8	8	8
2	4	3	3	4	1	8	1

Store bucket IDs for each pixel in an array that mirrors image's width and height. These bucket IDs are used to color at the end.

Euclidean Distance formula used to find most similar colors (lowest averages)

$$d = \frac{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2}{3}$$

r1, **g1**, **b1**: From palette or previous min. averages
r2, **g2**, **b2**: From current pixel color

Go through pixel by pixel...

RGB 93, 84, 95

Compare to palette colors or previous minimum averages using this formula

	34, 144, 164
	32, 100, 107
	2, 4, 9
	223, 240, 232
	209, 199, 173
	150, 118, 108
	184, 167, 137
	99, 75, 71

Minimum averages stored in array to be used in next pass

Pixel placed in bucket of most similar color to it

Repeat process until maximum number of passes is reached, or none of the minimum averages change

Applying contrast settings:

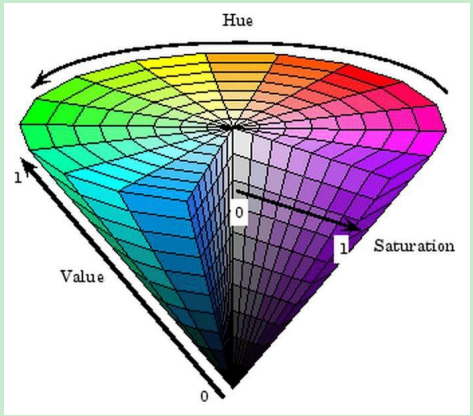
$$red = ((((\frac{r}{255.0}) - 0.5) * modifier) + 0.5) * 255.0;$$

(Repeat for pixel's green and blue values)

-**Contrast** is the lightness or darkness of different colors compared to each other

Applying tint and intensity settings:

-HSV (**h**ue **s**aturation **v**alue) color mode



Hue: Value from 0-360. Represents the "base color"

Saturation: Value from 0.0-1.0. Affects how much of the hue there is

Value/Brightness: Value from 0.0-1.0. 0.0 is completely black, while 1.0 is completely white.

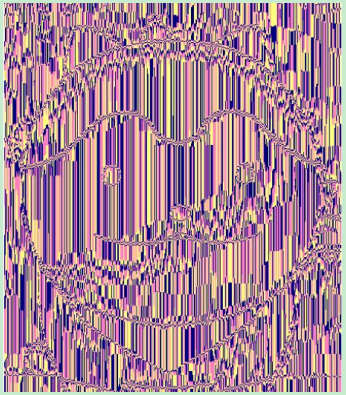
-Tint setting adjusts the brightness value of colors
    -**Tint** is the whiteness of a color
-Intensity setting adjusts the saturation value of colors
    -**Intensity** is the brightness or dullness of a color

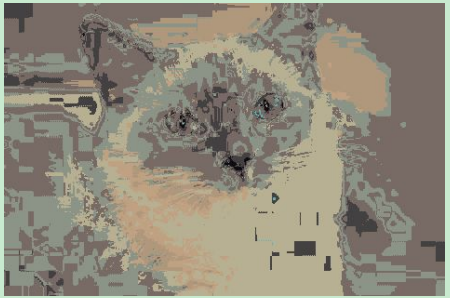
Demonstration

Development Process

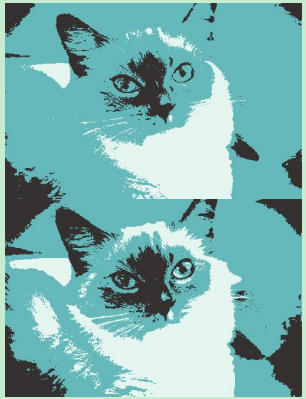
-BMVP & DCP
    -The K-Means algorithm, improving UI, making suggestions
-2021 graduate Colby Wall's blog
-Microsoft Learn
-Stack Overflow

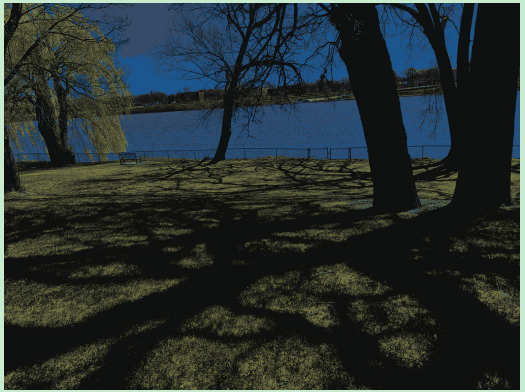
A visual timeline of progress...





A visual timeline of progress...



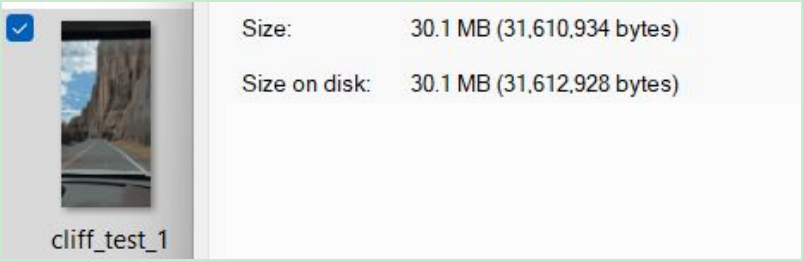


Exceptions

-Threading
-Pixel Smoothing + checking for isolated pixels
-Applying the contrast setting
-After running K-Means algorithm, letting user run additional passes from where it left off (saving the averages from the last pass)
-Contrast/tint/intensity settings set using sliders
-What if cartoon ends up being only one color?

Extensions

- Faster ways to process image
- Outlines
- More palette management options
- Reducing cartoon file size



Questions?