EE 440 Final Project Report

Professor Ming-ting Sun

TA: Maolong Tang

Department of Electrical & Computer Engineering

Chumei Yang

12/10/2018

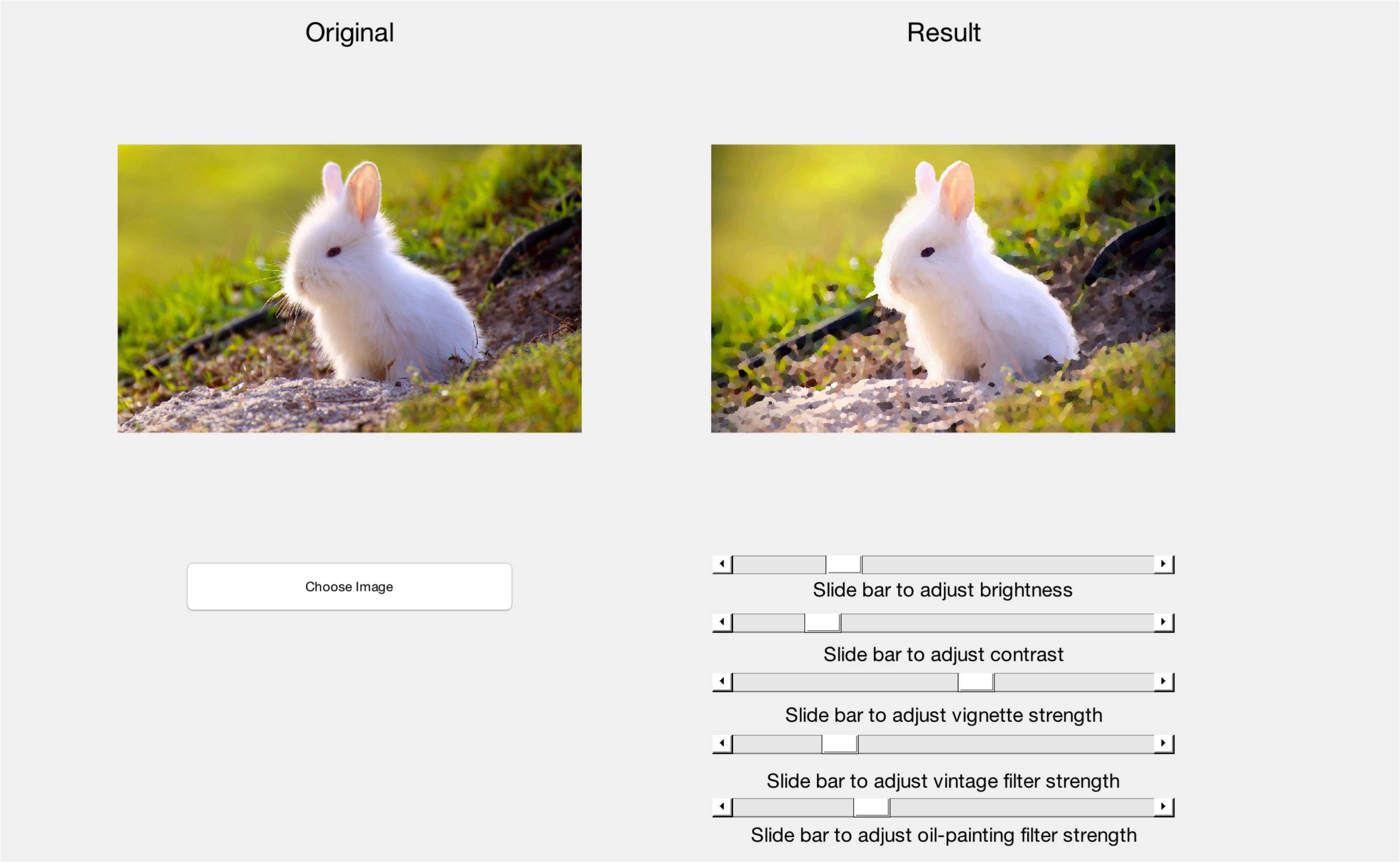
I. Abstract

For my final project for EE 440, I implemented a simple image processing app where one can adjust the brightness and contrast of the image, apply vintage filter, oil-painting filter, and vignette effect to an image. In the following sections of the report, I will list a few samples, explain how to use the app, and discuss what can be added later to achieve even better results.

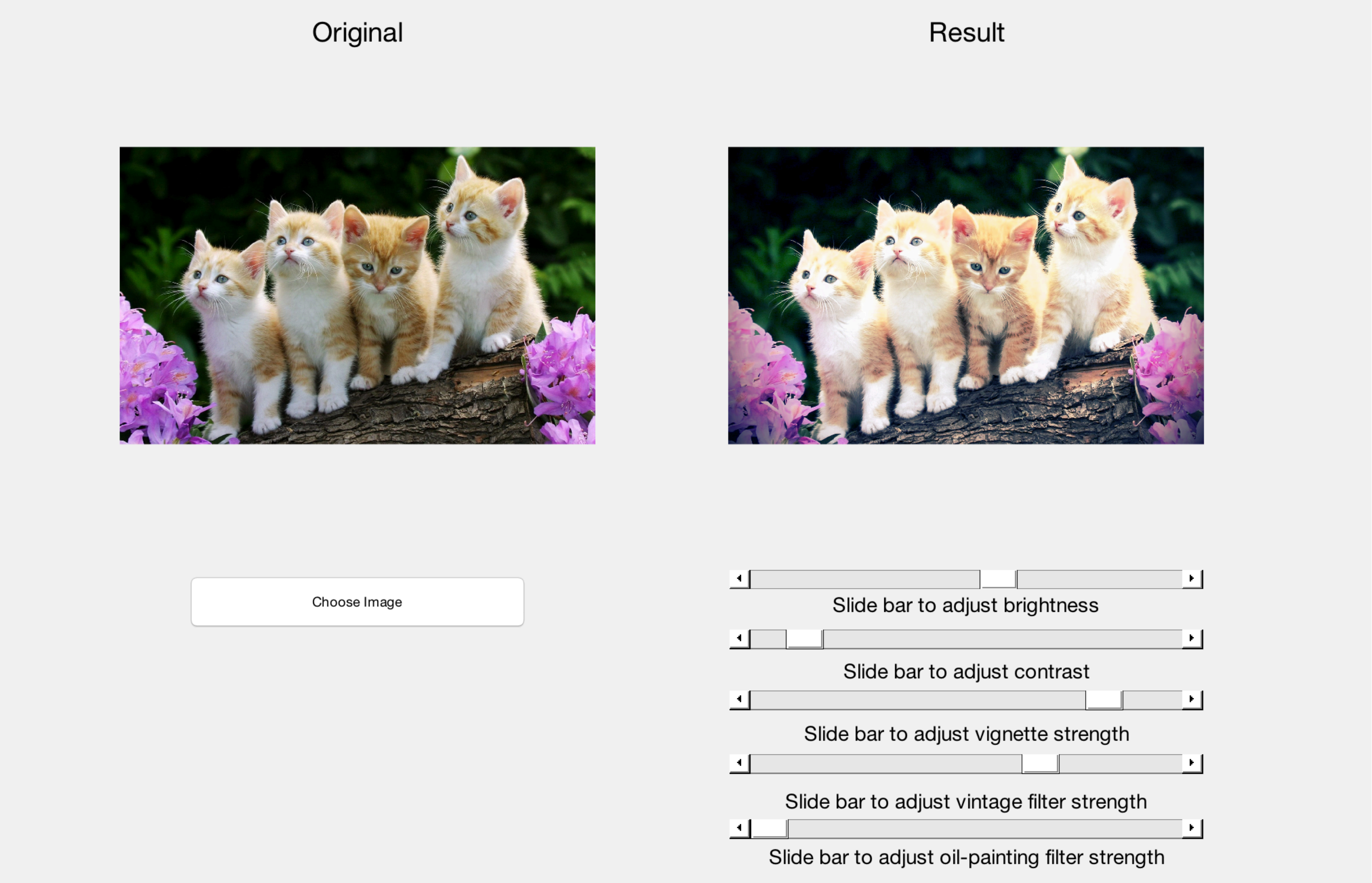
II. Sample results

Below are a few sample results. The original image is on the left, and the result is on the right with different effect strengths (strengths are shown under the image)

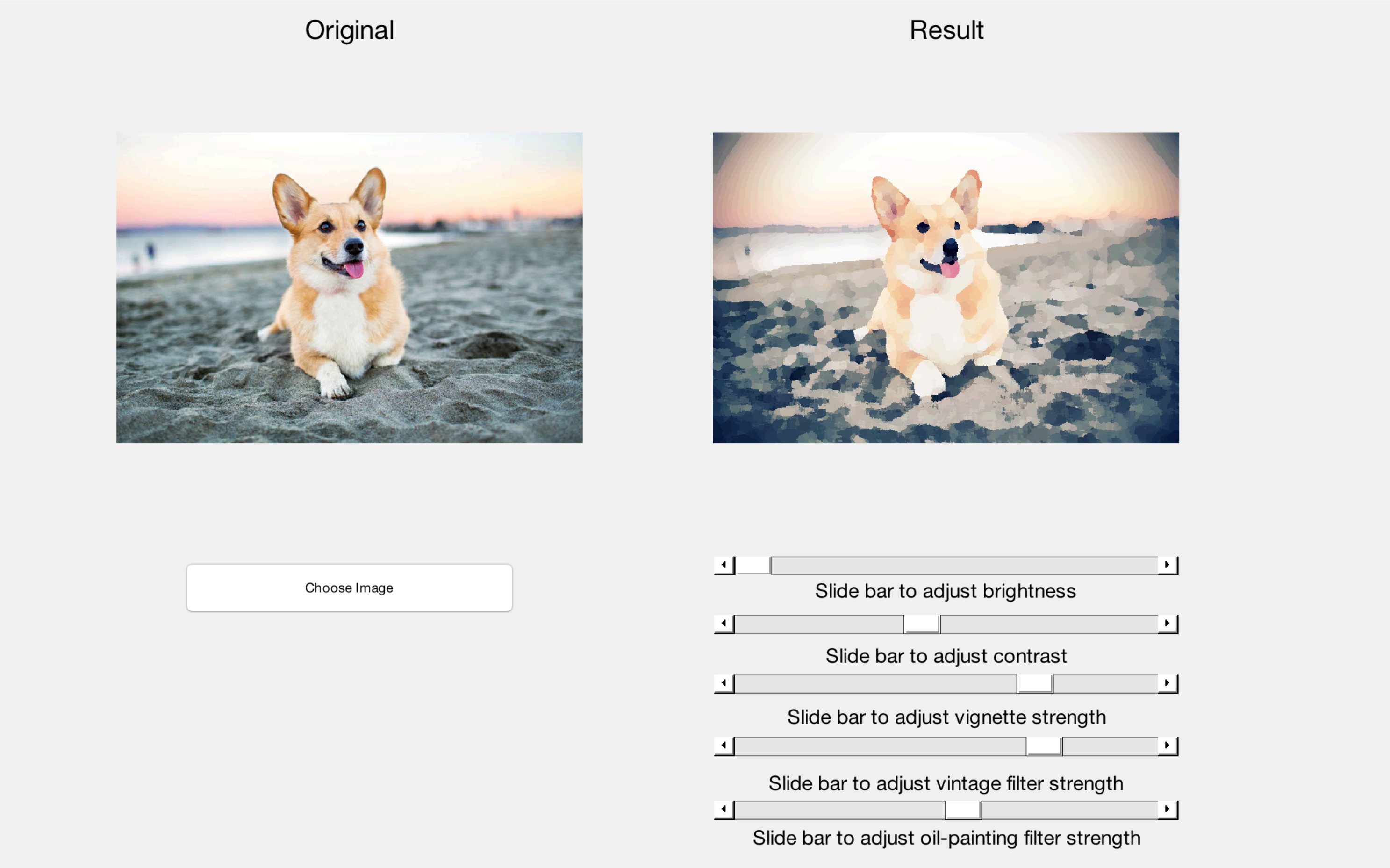
Sample 1:


Original
Result
Choose Image
Slide bar to adjust brightness
Slide bar to adjust contrast
Slide bar to adjust vignette strength
Slide bar to adjust vintage filter strength
Slide bar to adjust oil-painting filter strength

Sample 2:


Original
Result
Choose Image
Slide bar to adjust brightness
Slide bar to adjust contrast
Slide bar to adjust vignette strength
Slide bar to adjust vintage filter strength
Slide bar to adjust oil-painting filter strength

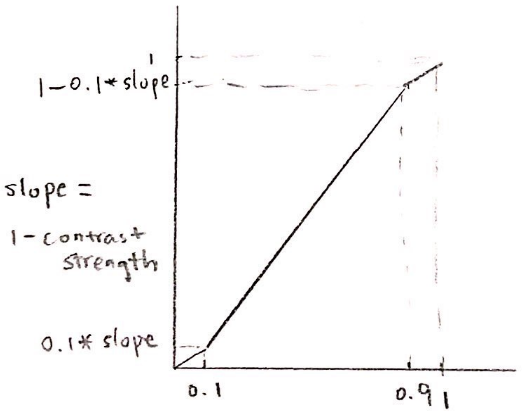
Sample 3:


Original
Result
Choose Image
Slide bar to adjust brightness
Slide bar to adjust contrast
Slide bar to adjust vignette strength
Slide bar to adjust vintage filter strength
Slide bar to adjust oil-painting filter strength

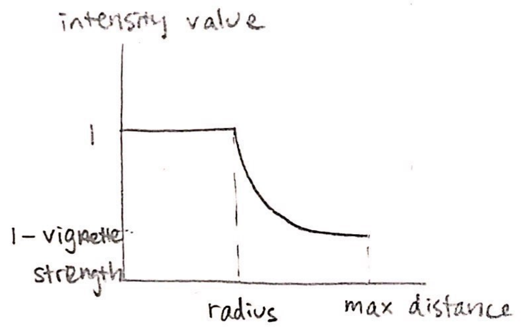
III. Effects and algorithms

I want to achieve 5 different effects. I will discuss each effect and how it is implemented below. I wrote all of the codes on my own and did not use any Matlab built in functions, but some ideas and concepts are from research.

1. Brightness. With the slide bar, a user can adjust the brightness of an image (only increasing though). In order to achieve this effect, I first converted the image to HSV and extracted the V channel, I then boosted the value according to its strength. If the value is greater than or equal to 1, I simply keep it as a 1.
2. Contrast. A user can adjust the contrast of an image with the slide bar. To achieve this effect, I first converted the image to HSV and extracted the V channel and then separated the intensity values into three ranges. The intensity for stretching contrast is set to 0.1-0.9. The slope for intensity range (0,0.1) and (0.9,1) is set to be (1 - contrast strength), and the values in between will undergo a linear transformation with slope greater than 1 based on the strength.



3. Vignette effect. A user can apply vignette effect to the image. The corner of the image will become darker to highlight the center of the image and give the image a vintage vibe. To achieve this, I first converted the image to HSV and extracted the V channel. Then I set a circle with a radius where the pixels inside the circle will remain unchanged. The intensity values of the pixels outside of the circle will undergo a function of (p(1)*dist^2+p(2)*dist+p(3))*value, a second-degree polynomial with coefficients determined by the strength specified by the user. If the pixel's distance to the center is radius, the scale factor will be 1. If the pixel is at the corner, i.e. it is at the maximum distance to the center, the scale factor will be (1 - vignette strength) (for example 0.5).



4. Vintage effect. A user can apply a vintage filter to the image. The image will have a more brown, golden tone. To achieve this effect, I first converted the image to HSV and and extracted the S (saturation channel). I lowered the saturation values based on the strength. I then extracted the R, G, B channels of the image and transformed the R and B channel. For R channel, the values below 128 will become even lower and the values above will become greater. For B, it is the opposite.
5. Oil painting effect. A user can apply an oil painting effect to the picture. An oil painting has more blocks compared to a digital picture, therefore, it has fewer intensity levels. To achieve this effect, I first specified the number of levels according to the strength. A stronger effect will have fewer intensity levels. I extracted the RGB channels. For each pixel, I examined the intensity levels of pixels around that pixel (10*10 square). For each pixel in that square, I get the average intensity of the RGB channels, determine its new level, and updated the counts of each level. The new level of the pixel will be the level with the most counts. I average the RGB values of the pixels in this square and apply this value to the pixel and reconstruct the image. The image now will be more blocky and thus have an oil painting effect.

IV. User instructions

To run this program, simply

1. Click on “main.m” and run it, and the program will open.
2. Click on “Choose Image” and choose the image to transform.
3. Slide bars to apply effects. (Note that each effect can only be edited once. After you are done with one effect and move on to another, this effected cannot be edited anymore)
4. The oil-painting effect may be a little slow, so please wait for a while.

V. Learning experience

I have learned a lot in doing this project. First of all, it is my first time writing a Matlab GUI, and I learned the basics in building a GUI. Second, I learned a lot of image processing concepts. I learned how to adjust each color channel to achieve different filters, and I learned how intensity levels can do to an image (both in RGB and HSV). There is so much more that can be added to this program, such as more filters, histogram equalization, more effects like cartoon, illumination, etc..

References to some concepts:

1. https://www.mathworks.com/matlabcentral/answers/264583-image-processing-for-the-vignetting
2. http://supercomputingblog.com/graphics/oil-painting-algorithm/