

CS101 Homework #3 Report

지준섭

1. Introduction

In this program, we need to make image processing program with python. The functions we need to implement are Lomo Effect, Contrast Adjust.

To adjust contrast, we need this equation:

$$\text{New color} = 128 + c \times (\text{old color} - 128)$$

And the constant c is determined by this equation:

$$c = \frac{(100 + \text{factor})}{100}, \text{ if } k < 0. \quad c = \frac{100}{(100 - k)}, \text{ if } k > 0.$$

To make lomo effect, we should apply this equation:

$$\text{new color} = \text{image color} - \frac{255 \% \text{ vigenette color}}{\text{intensity factor}}.$$

2. Algorithm

i) `main()`

Function `main` should show menus to the user and receive the input. To implement this, we should need following code:

```
quit = False
while not quit:
    user_input = raw_input("Please Select the Menu\nq)
Quit\ns) Show Image\nl) Lomo\na) Adjust Contrast\ninput> ")

    if user_input == 'q':
        quit = True
    elif user_input == 's':
        filename = raw_input("Enter the name of the file: ")
        show_img(filename)
    elif user_input == 'l':
        filename = raw_input("Enter the name of the file: ")
        lomo(filename)
    elif user_input == 'a':
        filename = raw_input("Enter the name of the file: ")
        k = int(raw_input("Enter the integer: "))
        adj_contrast(filename, k)
    else:
        print "Unknown Command. Try another input"
```

In this code, user will see the message like this:

Please Select the Menu

q) Quit
s) Show Image
l) Lomo
a) Adjust Contrast
input>

If user's input is 'q', then the code will make the variable 'quit' True. This will make the condition of 'while' False: the program would be ended. Else if input is s, the program will receive the name of the file, and call the function `show_img()`. If it's l, program will call `lomo()`, or if it's a, program will ask additional integer factor, and will call `adj_contrast()`. If the input is not q, s, l, or a, program will print 'Unknown Command. Try another input' and ask the menu again.

ii) `adj_contrast(img_fname, k)`

At first, we need to check the range of the factor k. Factor k should be (-99~99), so if k is out of this range, we should print error message and close the function by return.

If k is in proper range, we should adjust contrast to the photo with this code:

```
img = load_picture(img_fname)
w, h = img.size()
## Get size of the image

final = create_picture(w, h)
## Create new picture 'final', has the same size with img

for y in range(h):
    for x in range(w):
        r, g, b = img.get(x, y)

        if k < 0:
            c = (100.0 + k) / 100.0

        else:
            c = 100.0 / (100.0 - k)
        ## Calculate c, in case of c's sign

        r = int(128 + c * (r - 128))
        g = int(128 + c * (g - 128))
        b = int(128 + c * (b - 128))
        ## Calculate contrast values of each rgb values

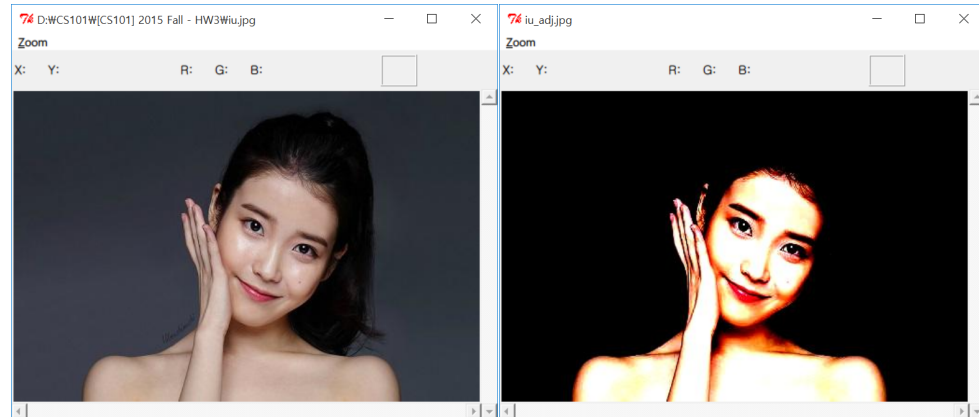
        final.set(x, y, (r, g, b))

final.save_as('iu_adj.jpg')
```

In this code, function gets the pixel's rgb value and calculate new contrast

value. The constant c is determined by factor k 's sign. New contrast values are set to new picture `final`'s pixel, and finally save the image `final` as 'iu_adj.jpg'.

This is the result with $k = 75$:



iii) `lomo(img_fname)`

`lomo` is a function that calls `addVignette` function twice with factor 2, 3 and save the new image file. Implementing this is simple:

```
img = load_picture(img_fname)
vig = load_picture("vignette1.jpg")

w, h = img.size()

final = addVignette(img, vig, w, h, 2)
final = addVignette(final, vig, w, h, 3)
## Call addVignette function twice

final.save_as('final_iu.jpg')
```

Just load the image `img` and `vig`, and adjust new vignette picture with factor 2 and 3 twice.

iv) `addVignette(inputPic, vignettePic, w, h, factor)`

`addVignette` is a function that changes the image's pixels into vignette values.

```
for y in range(h):
    for x in range(w):
        img_r, img_g, img_b = inputPic.get(x, y)
        vig_r, vig_g, vig_b = vignettePic.get(x, y)

        new_r, new_g, new_b = getNewColorValues((img_r, img_g,
img_b), (vig_r, vig_g, vig_b), factor)

        newimg.set(x, y, (new_r, new_g, new_b))

return newimg
```

This code will change all pixels' value into vignette value by calling `getNewColorValues`.

v) `getNewColorValues(inputPixel, vignettePixel, factor)`

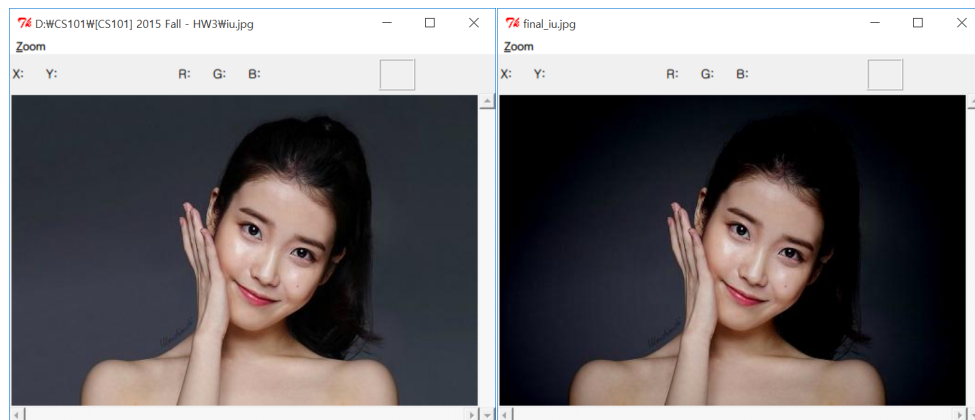
This function should return the rgb value that is applied vignette filter. To make this, all we have to do is just calculating each value in equation. The code below would do this job:

```
img_r, img_g, img_b = inputPixel
vig_r, vig_g, vig_b = vignettePixel

new_r = int((img_r - (255 % vig_r)) / float(factor))
new_g = int((img_g - (255 % vig_g)) / float(factor))
new_b = int((img_b - (255 % vig_b)) / float(factor))

return (new_r, new_g, new_b)
```

And this is the result:



3. Conclusion

I learned how to apply contrast and vignette filtering to images. It was very interesting to make image filtering software on my own. Now I feel I can implement almost all functions of the photoshop program. Now I want to add the functions of mosaic, blur filter. It will be implemented by processing the surrounding pixels of the pixel we want to change.