CSE 107: Lab 02: Simple Image Manipulations in Python.

<your name>
LAB: T 10:30-1:20pm
Yuxin Tian
September 12, 2022

Task 1: Computing the maximum value of an image. Rotating an image.

Figure 1: The Beginnings image rotated 90 degrees clockwise.

<Your answers to the task questions>

Task 2: Writing a function that computes the inverse of a grayscale image.

Figure 2: The inverse of the Watertower image.

<Your answers to the task questions>

Task 3: Creating a gradient grayscale image. Computing the image average.

Figure 3: The gradient image.

<Your answers to the task questions>

Task1.py

```
# Import pillow
from PIL import Image, ImageOps

# Import numpy
import numpy as np
from numpy import asarray

# Read the image from file.
im = Image.open('Beginnings.jpg')

# Show the image.
im.show()

# Convert image to gray scale.
im_gray = ImageOps.grayscale(im)

# Show the grayscale image.
im_gray.show()

<the rest of your code>
```

Task2.py

```
# Import pillow
from PIL import Image, ImageOps

# Import numpy
import numpy as np
from numpy import asarray

# Read the image from file.
im = Image.open('Watertower.tif')

# Show the image.
im.show()

# Print the image mode.
print("image mode is:", im.mode)

# Create numpy matrix to access the pixel values.
im_pixels = asarray(im)

# Import myImageInverse from myImageInverse
from MyImageFunctions import myImageInverse

im_inv_pixels = myImageInverse(im_pixels)

# Create an image from im_inv_pixels.
im_inv = Image.fromarray(np.uint8(im_inv_pixels))

# Show the inverse image.
im_inv.show()

# Save the inverse image to a file.
im_inv.save("Watertower_inv.tif")
```

```
# MyImageFunctions.py

# Import pillow
from PIL import Image, ImageOps

# Import numpy
import numpy as np
from numpy import asarray

def myImageInverse( inImage_pixels ):
    # This function takes as input a numpy matrix representing a grayscale image and
    # outputs another numpy matrix which is the image inverse of the input.
    # That is, for each pixel, output_value = 255 - input_value
    #
    # Syntax:
    #   out_numpy_matrix = myImageInverse( in_numpy_matrix )
    #
    # Input:
    #   in_numpy_matrix = the grayscale values of the input image
    #
    # Output:
    #   out_numpy_matrix = the grayscale of the inverse image
    #
    # History:
    #   S. Newsam   9/10/22     Created
    #   S. Newsam   9/11/22     Modified to use numpy instead of PIL image

<the rest of your code>
```

Task3.py

```
# Import pillow
from PIL import Image, ImageOps

# Import numpy
import numpy as np
from numpy import asarray

# The size of the gradient image.
rows = 100
cols = 256

# Create a numpy matrix of this size.
im_pixels = np.zeros(shape=(rows, cols))

<the rest of your code>
```