Q1: Playing with Frequencies [40 marks]

Develop an application for performing focus analysis on an image. Focus or lack thereof is an important tool employed by photographers to create aesthetically pleasing photographs. While we are far from discovering automated schemes for evaluating aesthetic scores of an image, we can definitely identify regions in an image that are in sharp focus. Your goal is to identify regions in an image that are in sharp focus. You can draw a rectangle on the image to identify regions of sharp focus as seen in Figure 1. Another option is to color each pixel according to the degree to which it is in focus.

Figure 1: Focus analysis

Your code should load a file and show regions are in sharp focus. You should use Python with OpenCV for this
purpose. So the program will work as follows:

```
$ python focus.py test-image.jpg
Focus analysis took 200 ms.
press any key to quit
```

The program will load and display the image with highlighted regions and wait for users input. The program will also print a message `Focus analysis took 200 ms.` that will tell us how long it took to perform focus analysis. The program will quit when a key is pressed.

**What to submit**

Please submit `focus.py` via Blackboard.

**Q2: Creating ASCII art [40 marks]**

Create a python program that converts a grayscale image into ASCII art. For example, I used the tool available at https://manytools.org/hacker-tools/convert-images-to-ascii-art/ to convert the image shown in Figure 1 into the ascii art image shown below (Figure 2).

![Figure 2: Ascii art](image)

Your code should work as follows:

```
$ python makeascii.py --width=80 focus.jpg output.txt
```

Here `width` specifies the width of the output in characters. I will leave it up to you to decide whether or not to use colors. You should make use of printable characters in the ASCII table listed here.

**What to submit**

Please submit `makeascii.py` via Blackboard.
Q3: Image Blending [20 marks]

Write a program that takes two images $I_1$ and $I_2$ and blends them to construct the resulting image $I$. The program will display the resulting image.

Specifically: $I = \alpha I_1 + (1 - \alpha) I_2$, where $\alpha$ lies between 0 and 1.

$python$ blend.py <alpha> <im1> <im2>
press any key to quit

We assume that the images are color (RGB). We also assume that the two input images can have different sizes (width and height). If the input images have different sizes then the size of the output image is equal to the size of the first image.

What to submit

Submit blend.py via Blacboard.

General comments

If you wish you can upload IPython Notebooks for the three questions in lieu of the python files. Please ensure that the notebooks clearly indicates where/how to specify the name of the input file.

Start early and best of luck.