

CS 8, Winter 2015
Homework Assignment #? (draft)

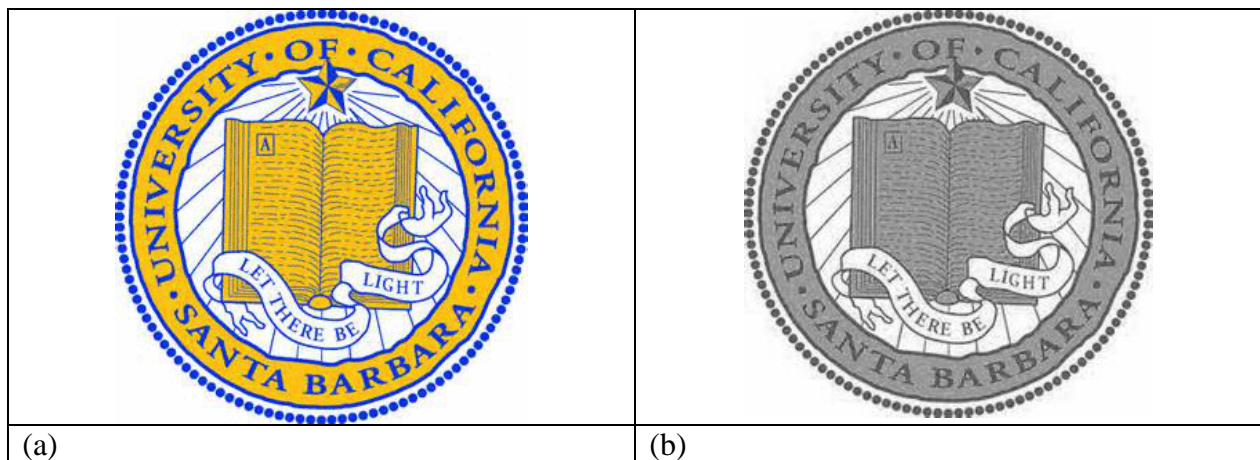
Assignment Overview

This assignment allows you to use Python cImage class to perform some simple image processing operations.

Assignment Description

An image filter is a mathematical operation, or an image to image transformation, that takes an image as the input and produces another image as the output. There are many operations that can be formulated this way, we will attempt the following:

- a) Given an input color image,
- b) **Grayscale**: take a color image and convert it to a grayscale image by averaging the three color channels red, green and blue values and set these channels to the same average value. If the input image is already a grayscale one, this operation has no effect,
- c) **Negative**: take a color or grayscale image and invert the values of every pixel to generate a negative image. If a negative image is given, the operation will generate a positive image,
- d) **Noise**: add random amounts of noise (either additive or multiplicative) into an image to simulate image corruption for a “salt-and-pepper” effect,
- e) **Average**: A filter operation to get rid of salt-and-pepper noise by replacing a pixel value by the average value of its neighbors, and
- f) **Median**: A filter operation to get rid of salt-and-pepper noise by replacing a pixel value by the median value of its neighbors.





(c)



(d)



(e)



(f)

Assignment Deliverables

The deliverable for this assignment is the following file:

im.py – the source code for your Python program

Be sure to use the specified file name and submit it for grading via the **turnin** system before the project deadline.

Assignment Notes:

1. The only function that is exposed (how your program will be tested) is

filter (inputFile, operation, param = {}, outputFile = "")

- inputFile: the input image file name
- operation: one of the following: grayscale, negative, noise, average, and median
- param: a dictionary of parameters for the operation, default is an empty dictionary
- outputFile: the output image filename, default is a null string (no output)

2. As different filter operations may require different sets of parameters, a mechanism must be used to create a common signature for the filter function. One such mechanism is to group all given parameters into a dictionary. Each entry in the dictionary is a parameter name with its value.
3. In particular, grayscale and negative filters take no parameters. Noise filter can take up to two optional parameters: 'Mag' and 'Type.' 'Type' is either 'additive' or 'multiplicative' (default is 'multiplicative') and 'Mag' is a fractional number in the range of 0 to 1 (default is 0.05).

Multiplicative noise is applied this way (as a percentage of the original pixel values):

$$\{r,g,b\}^{\text{output}} = \{r,g,b\}^{\text{input}} * (1 + \text{random}() - 0.5) * \text{'Mag'} * 255$$

And additive noise is applied this way (as an absolute amount):

$$\{r,g,b\}^{\text{output}} = \{r,g,b\}^{\text{input}} + (\text{random}() - 0.5) * \text{'Mag'} * 255$$

Both average and median filters take two optional parameters: 'FilterWidth' and 'FilterHeight' (both default to 3). A neighborhood of size FilterWidth by FilterHeight centered around the current pixel is used to compute either the average or median values.