CSci 4270 and 6270
Computational Vision,
Spring Semester, 2021
Lecture 05 Exercise
Due: Saturday, February 13, 2021 at 5 pm EST

Problem

The gradient magnitude of an image, I, is defined as

$$g(x,y) = \left[\left(\frac{\partial I}{\partial x}(x,y) \right)^2 + \left(\frac{\partial I}{\partial y}(x,y) \right)^2 \right]^{1/2}.$$

When we model the image as a continuous function this is clear and unambiguous. When we turn to a discrete implementation, things are more ambiguous. In particular, the Jupyter notebook distributed for class shows two different implementations of the partial derivative, one using just two values and one using six values in each "Sobel kernel". One immediate consequence of this is that the gradient magnitude values for the Sobel derivatives will be larger than for the first. This can be partially corrected by scaling down the Sobel results by a factor of 8 (think about why this is). But there will still be differences.

Write a short script that reads an image (as grayscale), computes the gradient magnitude using the two-valued kernel (the -1/2, 0, 1/2 kernel) and using (and rescaling) the Sobel kernels. It should then compare the gradients to determine

1. The average two-valued gradient magnitude (accurate to one decimal place).
2. The average Sobel gradient magnitude (accurate to one decimal place).
3. The average (absolute) difference betwen the two gradient magnitudes (accurate to one decimal place).
4. The maximum difference between the gradients for pixels where the two-valued gradient magnitude is larger (accurate to one decimal place)
5. The maximum difference between the gradients for pixels where the Sobel gradient magnitude is larger (accurate to one decimal place).
6. The percentage of pixels (accurate to the nearest integer) where the two-valued gradient magnitude is larger than the Sobel gradient magnitude.

Start from the code provided. Note: the second argument to both `cv2.Sobel` and `cv2.filter2D` should be `cv2.CV_32F`.