I400/B457 Intro to Computer Vision - Programming assignment #2

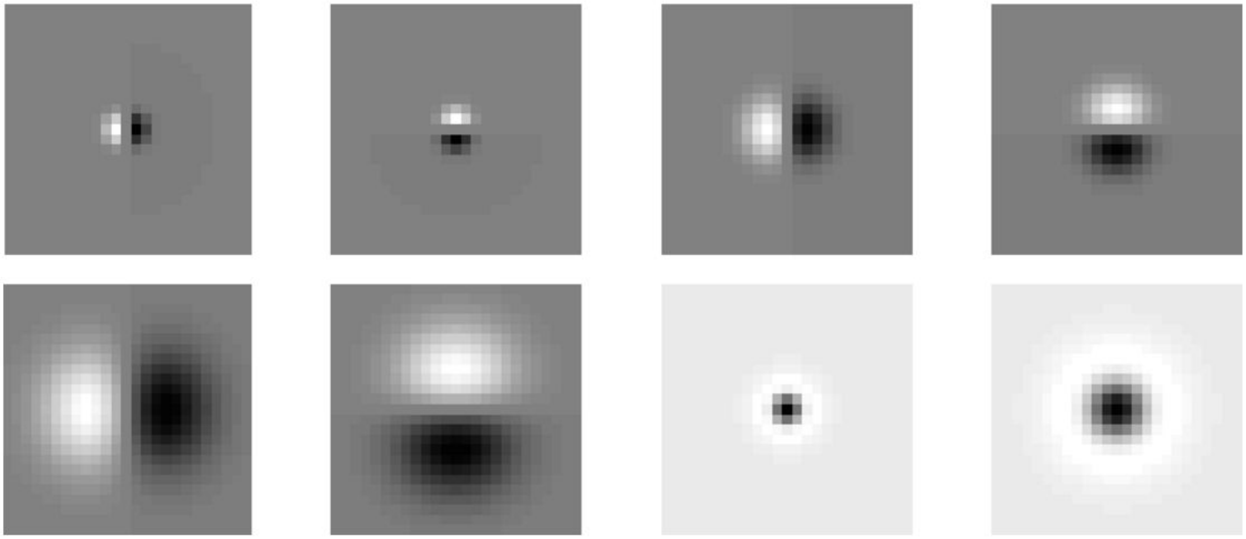
This assignment relies on lecture4_1_texture.pptx

Problem 1: Filter banks

0. Download “zebra.jpg” from Canvas.

1. [30%] Prepare filter banks for texture representation (slides 23-55 of lecture4_1_texture.ppt). Prepare 6 different “derivative of Gaussian” filters by convolving 2-D Gaussian filters with a gradient filter (e.g., Guassian*[1 -1]). More specifically, create 2-D Gaussian derivatives in the horizontal and vertical directions at 3 different scales σ = 2, 4, 8. Also create an additional center surround filter by taking the difference of two isotropic Gaussian functions at two different scales, i.e. G8(x, y) − G4(x, y) and G4(x, y) − G2(x, y).

They should (somewhat) look like this:



Use SciPy functions to do this:
http://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.ndimage.filters.gaussian_filter.html
http://docs.scipy.org/doc/scipy-0.15.1/reference/generated/scipy.signal.convolve2d.html

Notice that you will want to add the optional argument “`mode='same'`” for the `convolve2d.` Save the generated filters as “filter1.jpg”, ... , “filter8.jpg”.

2. [30%] Perform convolution using the created 8 different filters with the test images. Measure each pixel’s squared (i.e., x^2) filter response per filter. Save the results as “zebra_activation1.jpg”, … . Tip: Rescale the array values so that its max value becomes 255 before saving it as an image. A[x,y] = A[x,y]/max * 255.

3. [40%] Treat each pixel of the zebra image as a 8-D data, based on its squared filter responses (i.e., x^2).

Get the 8-D vector of the center pixel of the image. Next, iterate through 8-D vector of each pixel in the image, and compare it with the 8-D vector of the center pixel. Compute their Euclidean distance, and save them in a new array:

```
for each (x,y)
      D[x,y] = euclidean_distance(V[x,y], V[cx,cy])
```

where `V[x,y]` is actually an array (i.e., vector) with 8 values from the above step 2, and `(cx,cy)` is the center location.

Save the result array D as a grayscale image by rescaling it. Make the maximum value in D to become 255. Save the image as “zebra_texture_comparison.jpg”.