

Homework 3 solution template

CMPSCI 370 Spring 2019, UMass Amherst
Name: Subhransu Maji

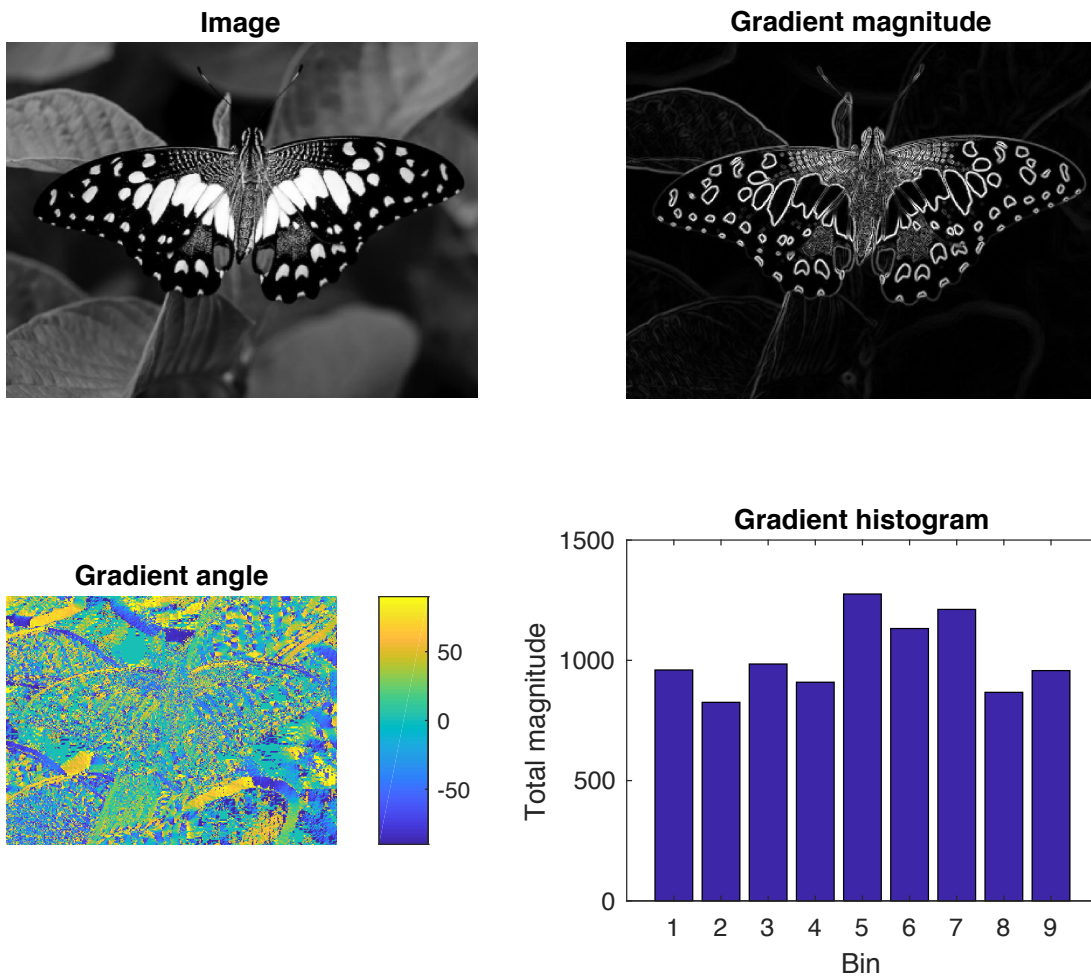
Here is a template that your solutions should roughly follow. Include outputs as figures, and code should be included in the end.

1 Image filtering [10 points]

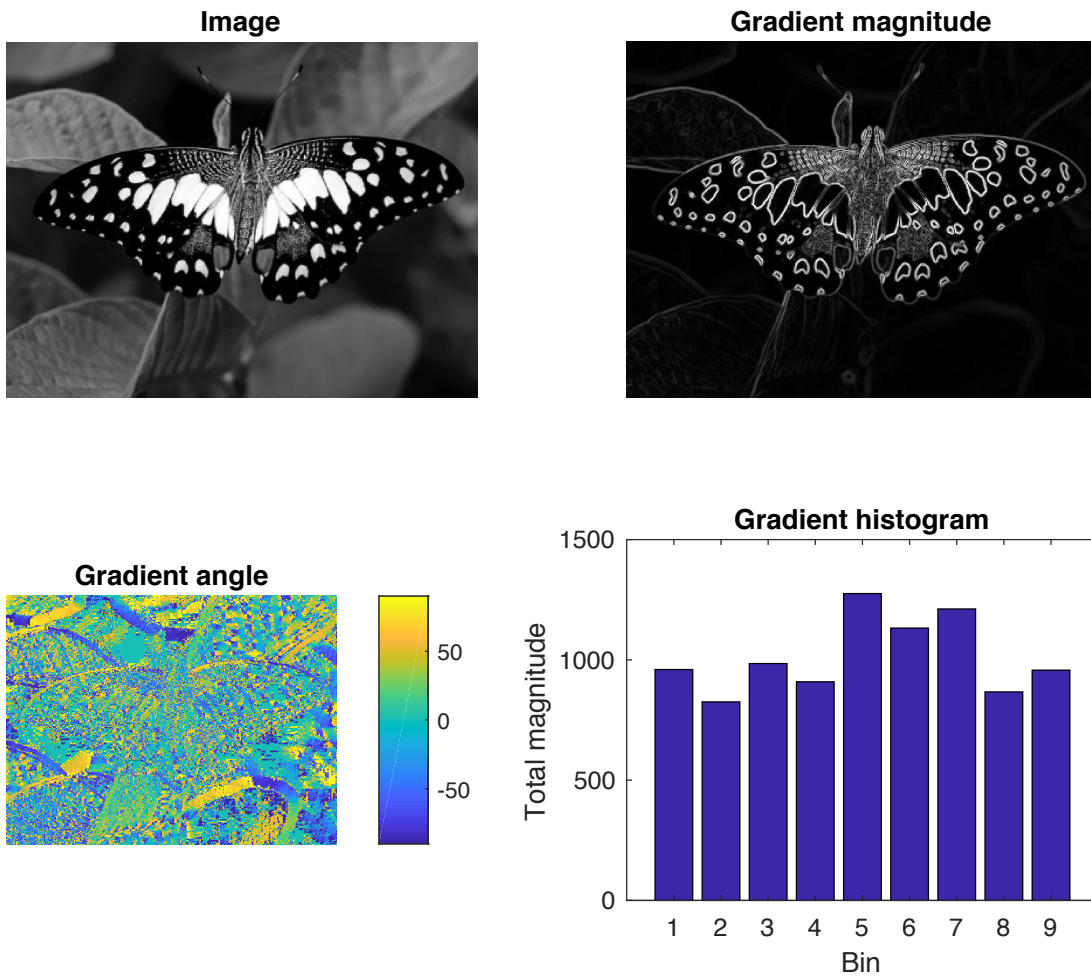
- Why is filtering with a Gaussian kernel preferable over a box kernel for denoising an image?
- What is the effect of increasing the σ of the Gaussian kernel on the result of filtering?
- When is median filtering preferable over Gaussian kernel filtering for denoising?
- Why is it a good idea to smooth an image before filtering with a derivative filter?
- What does filtering an image with a Laplacian of Gaussian filter do?

2 Image Gradient and Orientation Histogram

1. Visualizing gradient magnitude, angle and histogram



2. Visualizing gradient magnitude, angle and histogram for smoothed image



3 Corner detection

1. Output for the checkerboard image. When you get the correct output of corner detectors, the heatmaps of corner scores should have higher values (more yellowish) around corners and lower (more bluish) elsewhere.

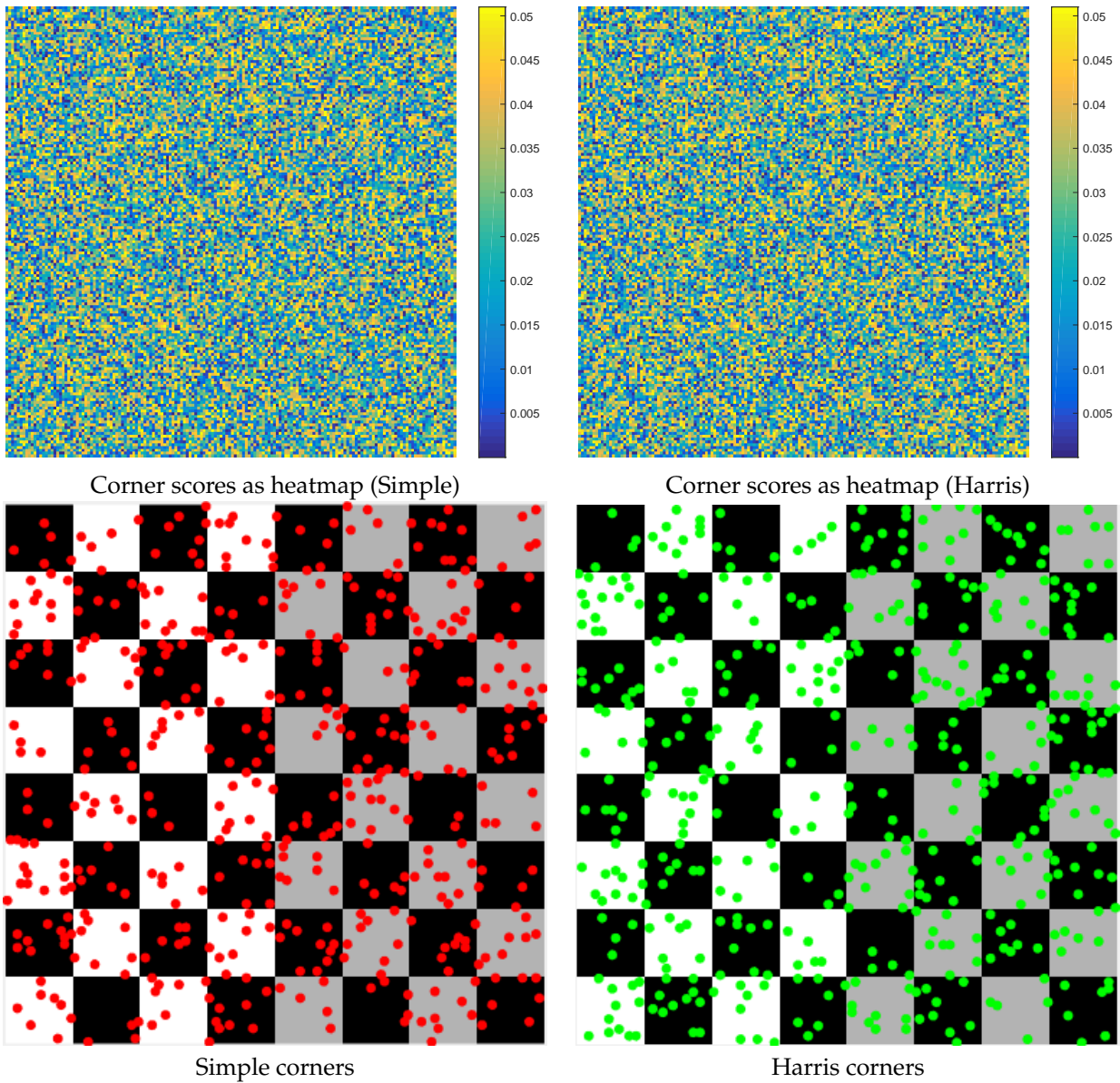


Figure 1: Results for the checkerboard image.

2. Output for the `'polymer-science-umass.jpg'` image

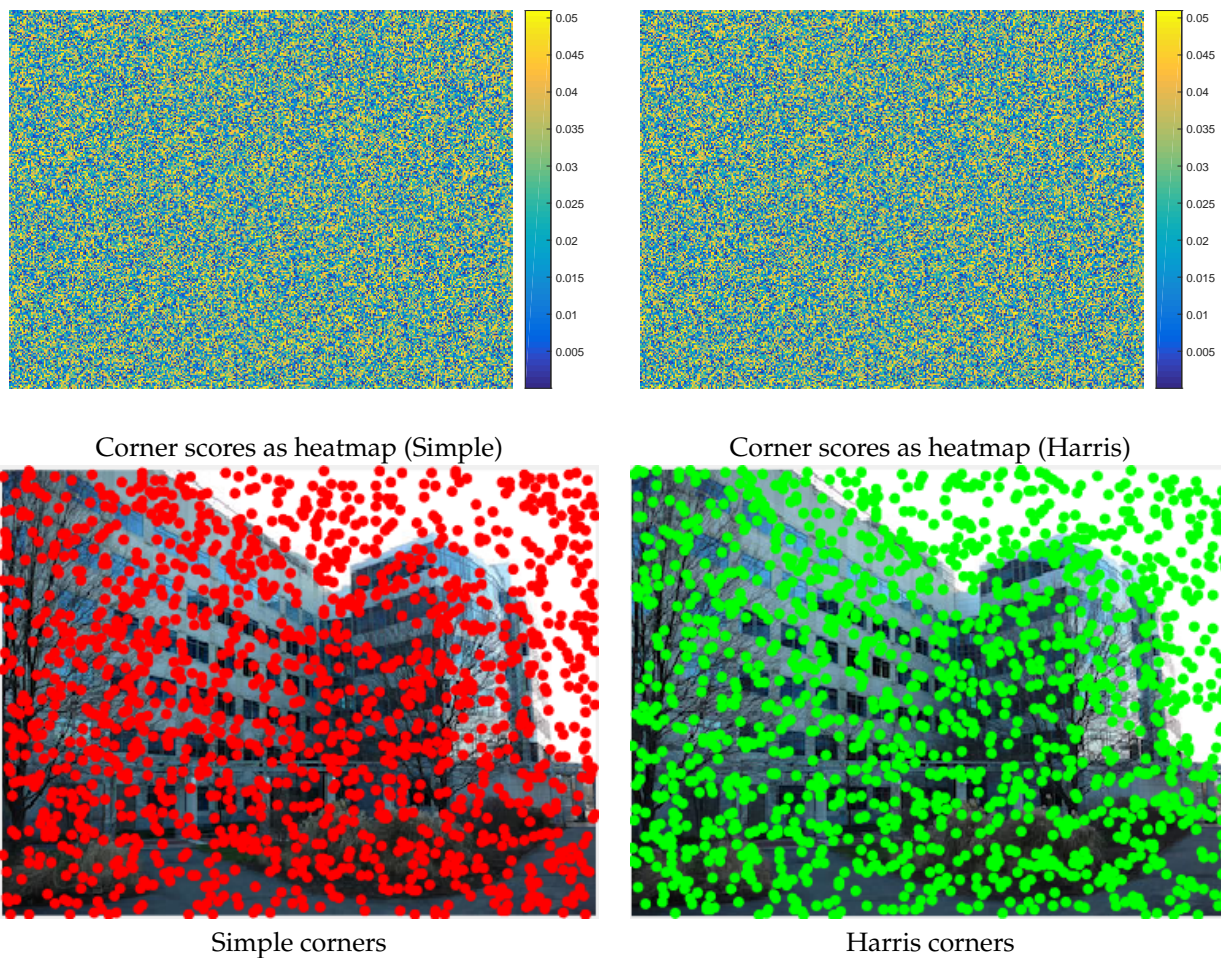
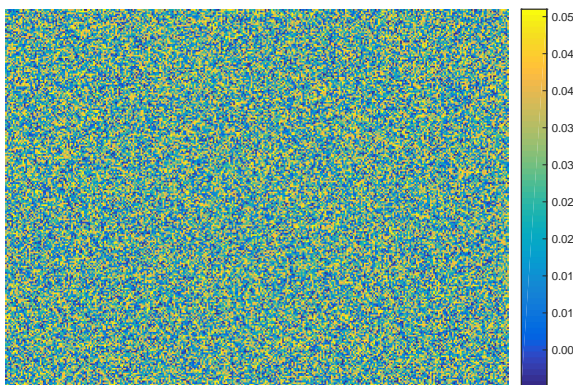


Figure 2: Results for the `'polymer-science-umass.jpg'` image.

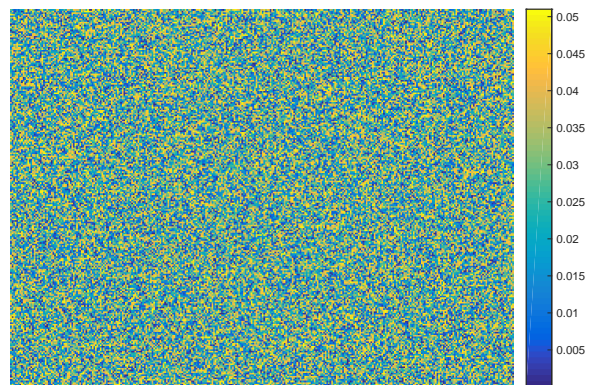
3. Output for the image of your own choice



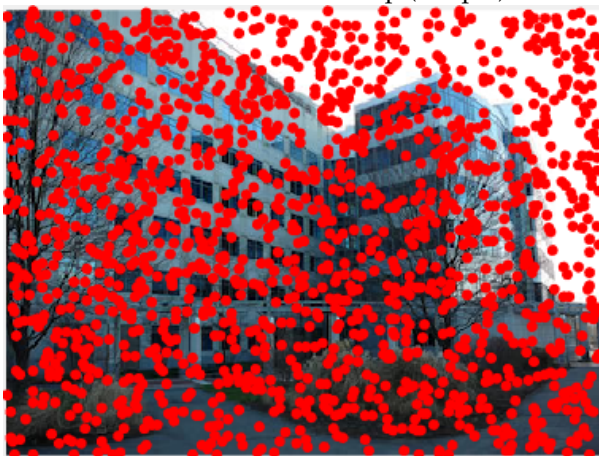
Input image



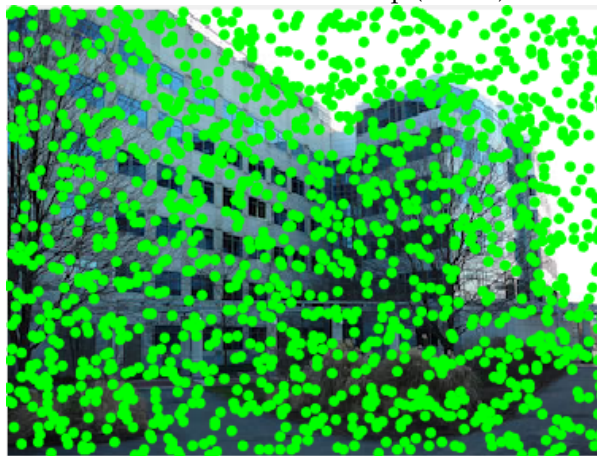
Corner scores as heatmap (Simple)



Corner scores as heatmap (Harris)



Simple corners



Harris corners

Figure 3: Results for the image of your own choice.

4 Solution code

Include the source code for your solutions as seen below (only the files you implemented are necessary). In latex the command `verbatiminput{alignChannels.m}` allows you to include the code verbatim as seen below. Regardless of how you do this the main requirement is that the included code is readable (use proper formatting, variable names, etc.) A screenshot of your code works to provided you include a link to source files.

4.a detectCorners.m

4.b imageGradient.m