CS283 Assignment Submission Guidelines

As part of the weekly assignments for this course, you are required to submit three types of work: typeset text and equations, Matlab code, and Matlab results. The purpose of this document is to provide formatting guidelines for these submissions—guidelines that are intended to ensure that each student's work is evaluated fairly. Submitted work that does not adhere to these guidelines may receive a zero grade.

How and when to submit

Homework is submitted electronically through submission to the appropriate *Assignment x* on the course website. Each submission will be a single ZIP archive that contains the files described below. Note that the submission system closes exactly at the deadline, so it is ill-advised to wait until the last minute. If you do miss the deadline, and if you have not already exhausted your allotted "late days", you may instead submit to the appropriate *Assignment x (late)*.

What to submit

Your ZIP archive will contain the following files. The main file is a single Matlab Live Script, which has file extension `.mlx`. This single file will include the typed text, latexed equations, embedded Matlab code, and Matlab results for your responses to all questions in the assignment. The file should include a header with your name and the assignment number, and there should be clear headers for each question and part.

For those familiar with Python, a Matlab live script is similar to an iPython notebook. To get started, read the vendor's online introduction[1], including the instructions for inserting latex equations[2].

In many cases, the executable code in your live script will require loading external data files, or Matlab functions from external m-files. The required external data and function files should be included in your ZIP archive, inside of sub-folders named `data` and `src`, respectively. You should verify that all code in your live script executes without error before submission, as files that cause errors may not be graded. If you are unable to create a complete working piece of code for a particular question, comment-out the parts that cause errors and provide a discussion of what you were trying to do so that we can give you partial credit.

Matlab coding guidelines

This is not a programming class, so programming style is not an explicit teaching objective. That said, code must be clearly-written and reasonably efficient, with comments to aid understanding. Here are a few suggestions to make sure your code receives full credit:

1. Begin each executable code cell with a *short* comment that provides the question number (if applicable) and a description of its function.

2. Provide *concise* comments throughout the code that describe its functionality.

3. Do not "hard-code" any constants. For example, if you need to create an all-black image of the same size as an input image `im` that happens to be 640×480, use `zeros(size(im))` instead of `zeros(640,480)`.

4. Avoid loops, and use parallel matrix and array operations instead. The commands `reshape`, `permute`, `find`, `sub2ind`, and `ind2sub` are useful for this.

For further information about writing clear and efficient Matlab code see Datatool's programming style guidelines[3] and Matlab's Techniques for Improving Performance[4].

[1] https://www.mathworks.com/help/matlab/live-scripts.html
[2] https://www.mathworks.com/help/matlab/matlab_prog/insert-equations.html
[3] http://www.datatool.com/downloads/matlab_style_guidelines.pdf
[4] http://www.mathworks.com/help/matlab/matlab_prog/techniques-for-improving-performance.html