ME565 - Winter 2021 **Due: 5:00:00pm Friday February 26^{th}, 2021**
Homework 4

Exercise 4-1: Load the image `recorder.jpg`. Convert to grayscale and compress the image using the FFT.

(a) Design a compression threshold to keep exactly 10% of the original Fourier coefficients. Compute the L_2 norm of the error between the new compressed image and the original image. Also compute the L_2 norm of the Fourier transformed versions of the compressed and original images.

(b) Repeat for a compression that only keeps 1% of the original Fourier coefficients.

Exercise 4-2: Now, we will use the FFT to simultaneously compress and re-master an audio file. Please download the file `r2112.mat` and load this file (`load r2112.mat;`) at the beginning of your code for this problem to load the audio data into the matrix `rush` and the sample rate `FS`.

(a) Listen to the audio signal (`>>sound(rush,FS);`). Compute the FFT of this audio signal.

(b) Compute the power spectral density vector. Plot this to see what the output looks like. Also plot the spectrogram using the same parameters as in lecture 17.

(c) Now, download `r2112noisy.mat` and load this file to initialize the variable `rushnoisy`. This signal is corrupted with high-frequency artifacts. Manually zero the last 3/4 of the Fourier components of this noisy signal (if `n=length(rushnoisy)`, then zero out all Fourier coefficients from `n/4:n`). Use this filtered frequency spectrum to reconstruct the clean audio signal. When reconstructing, be sure to take the real part of the inverse Fourier transform: `cleansignal=real(ifft(filteredcoefs));`.

Because we are only keeping the first 1/4 of the frequency data, you must multiply the reconstructed signal by 2 so that it has the correct normalized power. Be sure to use the `sound` command to listen to the pre- and post-filtered versions. Plot the power spectral density and spectrograms of the pre- and post-filtered signals.
