Access MATLAB by going to Resources - MATLAB and Toolboxes and follow the directions there.

- Do a basic implementation of JPEG:

1. Divide the image into non-overlapping 8x8 blocks.
2. Compute the DCT (discrete cosine transform) of each block. This is implemented in popular packages such as Matlab.
3. Quantize each block. You can do this using the tables in the video or simply divide each coefficient by N, round the result to the nearest integer, and multiply back by N. Try for different values of N.
4. You can also try preserving the 8 largest coefficients (out of the total of 8x8=64), and simply rounding them to the closest integer.
5. Visualize the results after inverting the quantization and the DCT.

- Repeat the above but instead of using the DCT, use the FFT (Fast Fourier Transform).
- Repeat the above JPEG-type compression but don't use any transform, simply perform quantization on the original image.
- Do JPEG now for color images. In Matlab, use the rgb2ycbcr command to convert the Red-Green-Blue image to a Lumina and Chroma one; then perform the JPEG-style compression on each one of the three channels independently. After inverting the compression, invert the color transform and visualize the result. While keeping the compression ratio constant for the Y channel, increase the compression of the two chrominance channels and observe the results.
- Compute the histogram of a given image and of its prediction errors. If the

1. pixel being processed is at coordinate (0,0), consider
2. predicting based on just the pixel at (-1,0);
3. predicting based on just the pixel at (0,1);
4. predicting based on the average of the pixels at (-1,0), (-1,1), and (0,1).

- Compute the entropy for each one of the predictors in the previous exercise. Which predictor will compress better?

Mark as completed