

## EECS6323 - Assignment 9: DCT-based compression à la JPEG

Assigned: Mar 21 (Tu)

Due: Apr 1 (Mon) by 11.59 pm (A little extra time)

Percentage of total grade: 10%

### Objective

1. To understand DCT-based compression
2. To see the impact of UV subsampling on full RGB reconstruction.

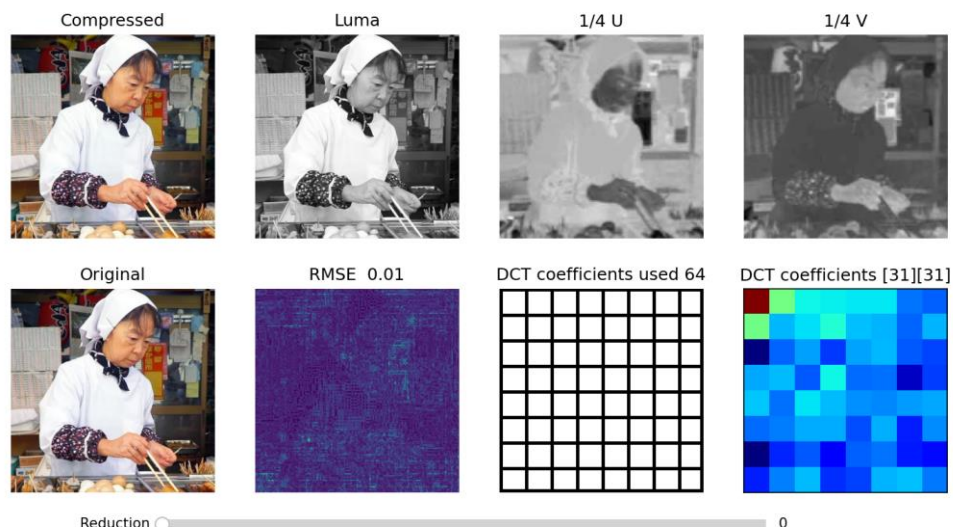
### Provided to you.

1. Two test images.
2. Video

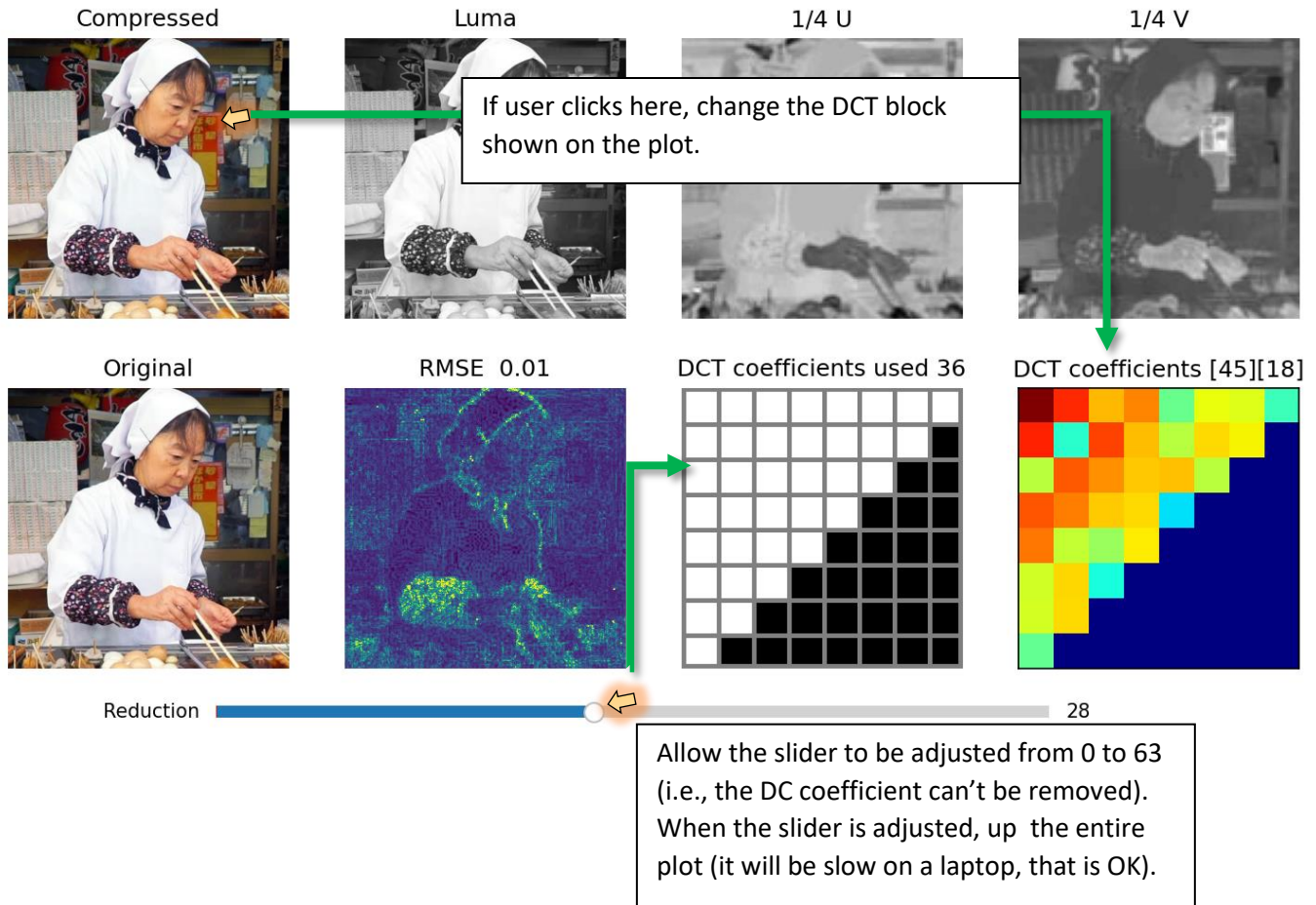
### How the program should work.

1. Open one of the test images and convert it to YUV.
2. Downscale UV by a factor of 4 (0.25 of its original size).  
-- Consider using scipy's "zoom" function
3. Perform 2D DCT on the 8x8 blocks as described in the notes. Do this for the Y', downsampled U, and V images. (you do not need to subtract -128 as JPEG does).
4. Based on the reduction factor, set certain DCT coefficients to 0. The reduction would start at the highest frequency DCT coefficient and work backward based on JPEG's zigzag ordering.
5. Reconstruct the new Y' and UV images based on the reduced DCT values. Note that a reduction of 0 means no change to DCT values. At high reduction factors, you may encounter a few saturated pixels in the reconstruction, but that is fine (see video).
6. Upsample the UV images by a factor of 4 (again, consider using scipy's zoom()).
7. Compute the per-pixel RMSE error with the original image. Also compute the whole image RMSE.
8. Plot the following on two rows:  
Row 1: Reconstructed (i.e., compressed) (1) RGB image, (2) Y', (3) U, (4) V  
Row 2: (1) Original image, (2) RMSE image, (3) DCT coefficient reduction map, and (3) the coefficients from some DCT block [see point 10 - default is DCT block 31,31]  
\* For the DCT block coefficients, my plot is the log of the values.
9. Allow the user to adjust the reduction factor using a "slider."
10. If the user clicks on the RGB reconstruction image, use the clicked (x,y) position to select a different DCT block to show (see video).

### Plot example



### Example after user interaction



### What to submit via e-class:

Two files

1. Your Python code (zip)
2. Video of your program running w/o audio (mp4).

### Comments:

1. The assignment must be done in Python.
2. You are welcome to use ChatGPT.
3. Useful Python packages/functions:

zoom from `scipy.ndimage`  
dct and idct from `scipy.fft`

See Lecture 17 for the zigzag order table.