COC202 Computer Vision
Lab 11 – Image compression/compressed domain retrieval

In this lab, you will implement part of a JPEG codec, and then use DCT features to build a colour histogram in the compressed domain for image retrieval.

If you have not yet finished the exercises from the previous lab, do them first.

1. Write a function that is passed a colour image (with image dimensions multiples of 8). The function should perform part of the JPEG compression algorithm.

   In particular, the function should:
   (a) be passed an image and the q-factor (in [1;99]) as parameters;
   (b) convert the image to YC_bC_r space, then compress each channel separately;
   (c) (we skip the downsampling that is usually performed on the chrominance channels;)
   (d) (we also skip the level shift that is usually applied prior to DCT; for the purpose of these exercises it doesn't matter (much) whether the data is in [0;255] or [-128;127] while for Exercise 3 it will be more convenient to have positive data;)
   (e) process each channel in 8x8 blocks and apply DCT to each block;
   (f) quantise the DCT coefficients using the two quantisation matrices provided (see the .m file on learn; there is one table each for luminance and chrominance channels) and the q-factor (see code further below);
   (g) round all DCT coefficients;
   (h) return a matrix that contains all (rounded) DCT data which now corresponds to the compressed data for the image (it's not really compressed as we skip the following entropy coding stages).

   The following code can be used to obtain the scaling factor for the quantisation matrices from the q-factor:

```
% get scaling of quantisation matrix from q-factor (qf in [1;99])
if (qf < 50)
    q_scale = floor(5000/qf)/100;
else
    q_scale = (200 - 2*qf)/100;
end
```

   Matlab functions that should be useful here: rgb2ycbcr, dct2.

2. Write a function that, passed the DCT data generated in Exercise 1 and the q-factor, decodes and returns the image, i.e. performs the various (inverse) tasks of Exercise 1 in reverse order.

   Use both functions to compress and decompress an image – compare the original and compressed images to confirm that the compression was lossy.

   Using the q-factor, scale the quantisation tables to steer the compression and hence image quality.

   Matlab functions that should be useful here: ycbcr2rgb, idct2.

Once you have finished all exercises you may leave the lab.

Additional exercises for further study:

3. Perform compressed-domain retrieval on the image database from the CBIR lab. In particular, calculate colour histograms directly from DCT DC coefficients to achieve this.

   Hint: The easiest way to do this is to use the function from Exercise 1 (say with a q-factor of 50) to obtain the DCT data for an image, then extract only the DC coefficients (i.e. the top left coefficient of each 8x8 DCT block) to build a "DC image". The values in that DC image can then be scaled by a factor of 2 (to arrive at a range more useful for the subsequent histogram generation) and the result directly passed to the colour histogram function you wrote for the CBIR lab.

   Hint 2: The images from the CBIR lab do not have image dimensions that are multiples of 8, so you'll first have to (automatically) cut them to the right dimensions.