IMAGE COMPRESSION

- Discrete Cosine Transform

BY: Group 31

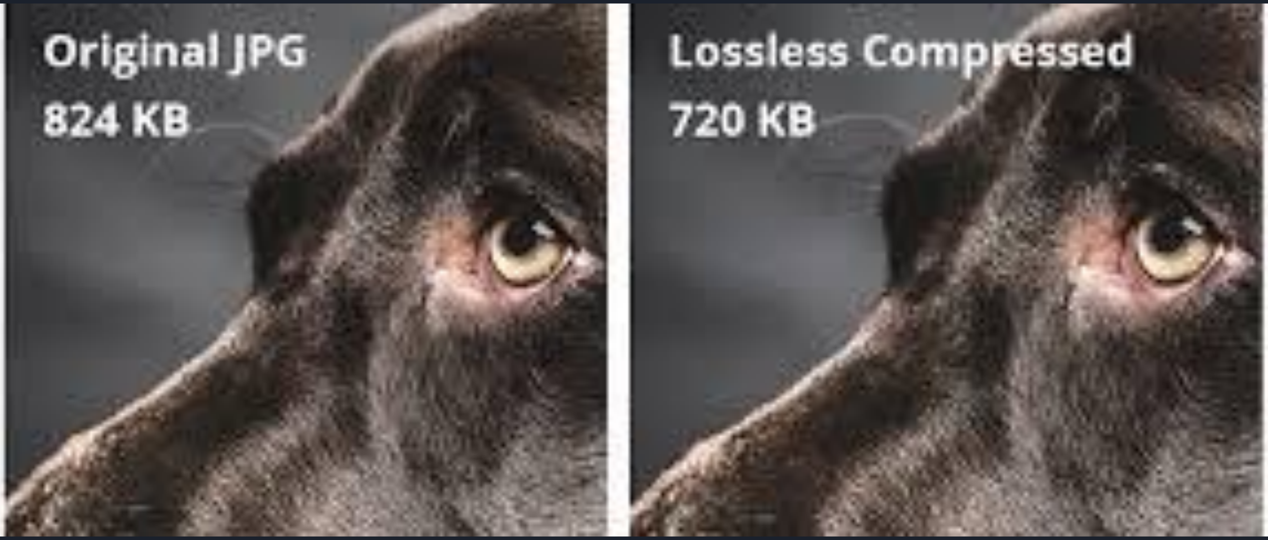
- Pradeeshwar J
- Dometti Sree Gnanesh
- Yacha Venkata Rakesh

Ref: https://www.math.cuhk.edu.hk/~lmlui/dct.pdf

IMAGE COMPRESSION

- Reduces Amount of data required to store an image.
- Useful when storing or transmitting data.
- Measured by compression ratio - Ratio of space needed to store after and before compression.
- Lossy compression provides higher compression ratio by sacrificing negligible data. Uses: Photography images can be converted to JPEG and stored thus occupying less space.
- Lossless compression on image has very less compression ratio but does not lose any data. Uses: Medical images need high precision in order to prevent any error.


Original JPG
824 KB
50% Lossy Compression
76 KB
80% Lossy Compression
38 KB


Original JPG
824 KB
Lossless Compressed
720 KB

DISCRETE COSINE TRANSFORM

- Algorithm used in most image compressions like JPEG.
- Image is first broken into 8x8 block pixels.
- From left to right, bottom to top, DCT is applied to each block.
- Each block is compressed using quantization.
- Array of compressed blocks is stored reducing storage space required.
- Can be decompressed by using Inverse Discrete Cosine Transform(IDCT) whenever necessary.

METHOD 1: EQUATION

- We use 4 for loops to compute the required compressed matrix .
- Since 8x8 matrix is usually used in JPEG, N = 8.

$$D(i,j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N} \right] \cos\left[\frac{(2y+1)j\pi}{2N} \right]$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } u = 0 \\ 1 & \text{if } u > 0 \end{cases}$$

METHOD 2: MATRIX MANIPULATION

Constants used:

- T , calculated for a specific DCT matrix.
- Q50 , A quality matrix playing major role in maintaining the compression. The more the level of quality matrix, the more the compression.
- Each element in the Image has from 0-255. Since DCT can be applied for -128 to 127 we subtract 128 on each element.

$$T = \begin{bmatrix} .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 & .3536 \\ .4904 & .4157 & .2778 & .0975 & -.0975 & -.2778 & -.4157 & -.4904 \\ .4619 & .1913 & -.1913 & -.4619 & -.4619 & -.1913 & .1913 & .4619 \\ .4157 & -.0975 & -.4904 & -.2778 & .2778 & .4904 & .0975 & -.4157 \\ .3536 & -.3536 & -.3536 & .3536 & .3536 & -.3536 & -.3536 & .3536 \\ .2778 & -.4904 & .0975 & .4157 & -.4157 & -.0975 & .4904 & -.2778 \\ .1913 & -.4619 & .4619 & -.1913 & -.1913 & .4619 & -.4619 & .1913 \\ .0975 & -.2778 & .4157 & -.4904 & .4904 & -.4157 & .2778 & -.0975 \end{bmatrix}$$

$$Q_{50} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

STEP 1: APPLYING DCT ON EACH 8X8 BLOCK

- We find D matrix by:

$$D = T\,M\,T'$$

Where T is transpose of matrix T.

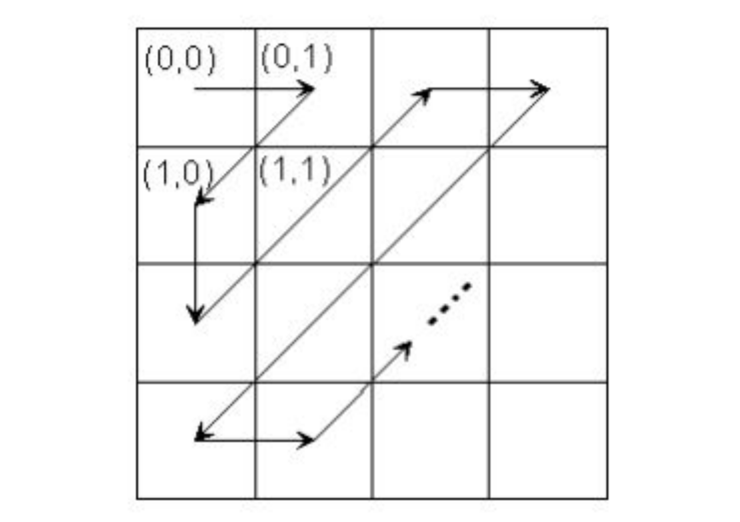
STEP 2: QUANTIZATION

- We divide each element of the D matrix by each element in the Q matrix, acquiring ‘C’.
    - We then round off each of the floating point numbers in C to get integer values.
    - Note that most of the lower triangle of the matrix becomes 0.

$$C_{i,j} = round\left(\frac{D_{i,j}}{Q_{i,j}}\right)$$

STEP 3: CODING

We then take 16(as it has a good data retaining - compression ratio) values in the following order and convert it into a 1D matrix.



DECOMPRESSION

We retrace the methods we used:

- Decode the 1D matrix to get the 2D matrix according to the figure in the previous slide and fill the rest with zeroes.
- Multiply element-wise with Quality matrix(let the output here be R)
- Apply N = T’ R T.
- We round off N and add 128 .
- Thus we obtain the original image with negligible data loss