INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR
Department of Electronics & Electrical Communication Engineering
Vision and Intelligent Systems (VIS)
EC69505 – Image Processing Laboratory

IMAGE AND VIDEO PROCESSING LABORATORY

IP MINI PROJECT

JPEG Compression

By : Group 40
Anik Mandal – 19EC39041
Swagata Naskar – 19EC39038

Objective :

- *Topic- JPEG Compression*
- *Entire JPEG pipeline must be designed.*
- *Code should be in Python/ Cpp*
- *Code should work for Colored Images for all sizes i.e. variable height and width.*

Introduction:

Image Compression:

Image compression addresses the problem of reducing the amount of data required to represent a digital image. It is a process intended to yield a compact representation of an image, thereby reducing the image storage/transmission requirements. Compression is achieved by the removal of one or more of the three basic data redundancies, Coding Redundancy, Interpixel Redundancy, Psychovisual Redundancy .Coding redundancy is present when less than optimal code words are used.

Interpixel redundancy results from correlations between the pixels of an image. Psychovisual redundancy is due to data that is ignored by the human visual system (i.e. visually non essential information). Image compression techniques reduce the number of bits required to represent an image by taking advantage of these redundancies.

An inverse process called decompression (decoding) is applied to the compressed data to get the reconstructed image. The objective of compression is to reduce the number of bits as much as possible, while keeping the resolution and the visual quality of the reconstructed image as close to the original image as possible. Image compression systems are composed of two distinct structural blocks: an encoder and a decoder.

Why we can compress image?

- *Statistical redundancy:*
  - *Spatial correlation:*
    - *Local – Pixels at neighboring locations have similar intensities.*
    - *Global - Reoccurring patterns.*
  - *Spectral correlation – between color planes.*
  - *Temporal correlation – between consecutive frames.*

- *Tolerance to fidelity:*
  - *Perceptual redundancy.*
  - *Limitation of rendering hardware.*

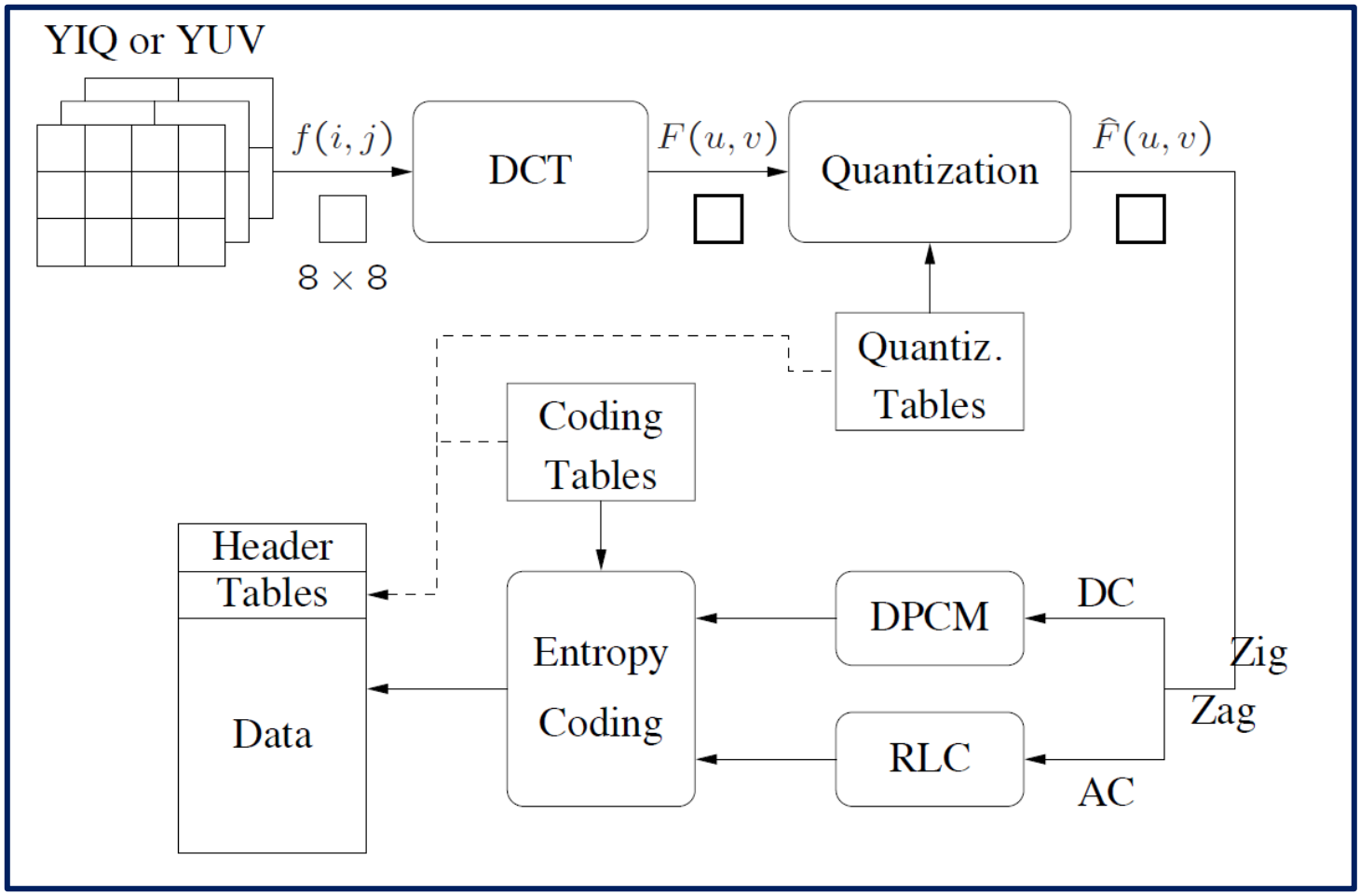


Fig : Block diagram for JPEG encoder.

DCT TRANSFORMATION

The most popular technique for image compression, over the past several years, was Discrete cosine transform (DCT). Its selection as the standard for JPEG is One of the major reasons for its popularity.

DCT is used by many Non-analytical applications such as image processing and signal-processing DSP applications such as video conferencing. The DCT is used in transformation for data compression. DCT is an orthogonal transform, which has a fixed set of basis function. DCT is used to map an image space into a frequency.

1-D DISCRETE COSINE TRANSFORM
DCT

$$C(u) = a(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{(2x+1)u\pi}{2N}\right]$$

$$u = 0,1,\ldots,N-1$$

$$a(u) = \begin{cases} \sqrt{\frac{1}{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u = 1,\ldots,N-1 \end{cases}$$

DCT has many advantages:

- *It has the ability to pack energy in the lower frequencies for image data.*
- *It has the ability to reduce the blocking artefact effect and this effect results from the boundaries between sub-images become visible.*

Algorithm:

- *An RGB to YCbCr color space conversion (color specification)*
- *Original image is divided into blocks of 8 x 8.*
- *The pixel values within each block range from[-128 to 127] but pixel values of a black and white image range from [0-255] so, each block is shifted from[0-255]to [-128 to 127].*
- *The DCT works from left to right, top to bottom thereby it is applied to each block.*
- *Each block is compressed through quantization.*
- *Quantized matrix is entropy encoded.*
- *Compressed image is reconstructed through reverse process. This process uses the inverse Discrete Cosine Transform (IDCT).*

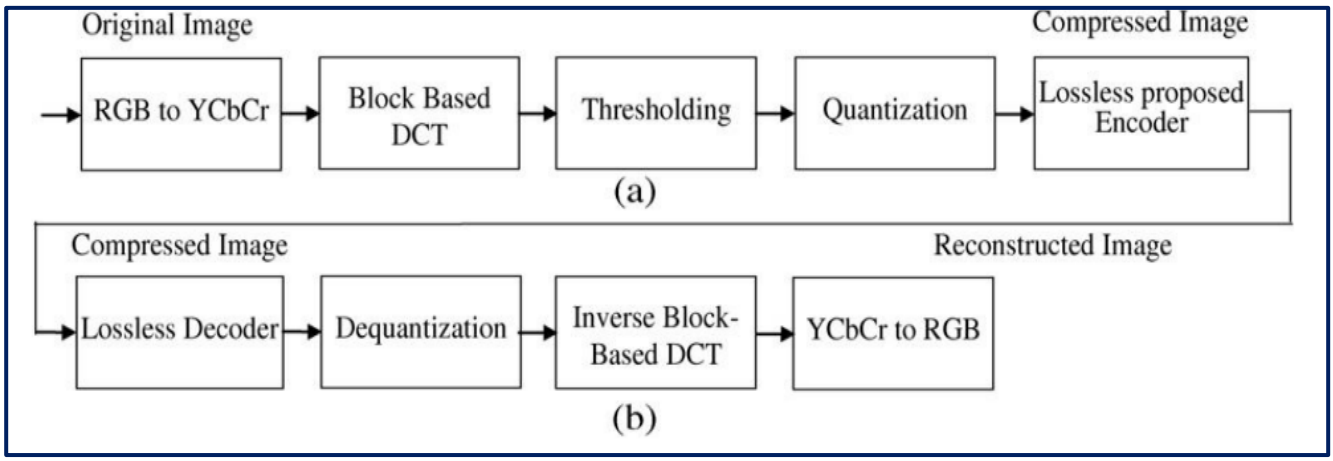


Figure . Compression algorithm scheme:
(a) compression step and (b) decompression step

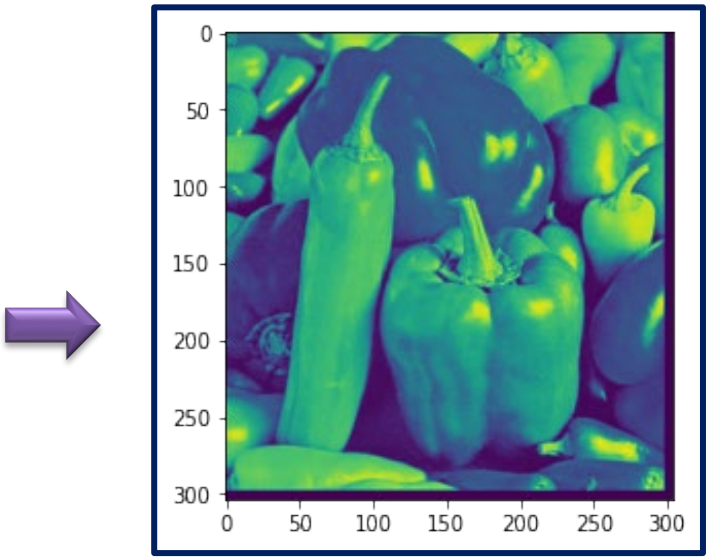
The 8*8 block for the following image:

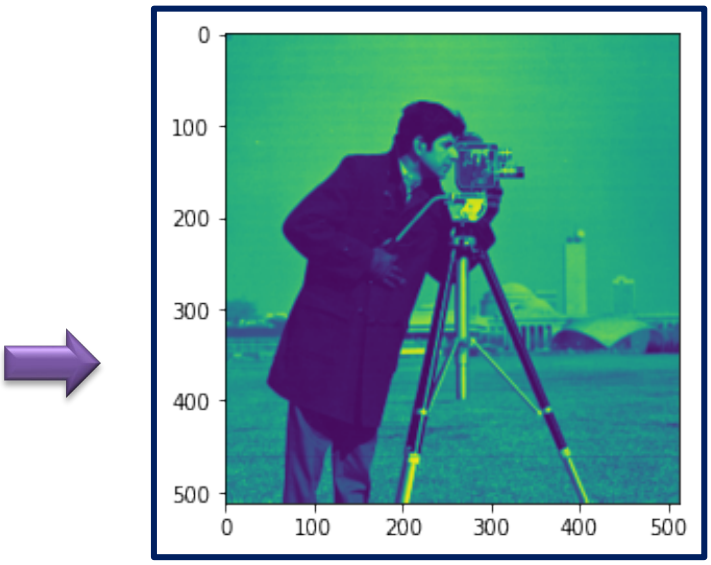
chrominance quantization table

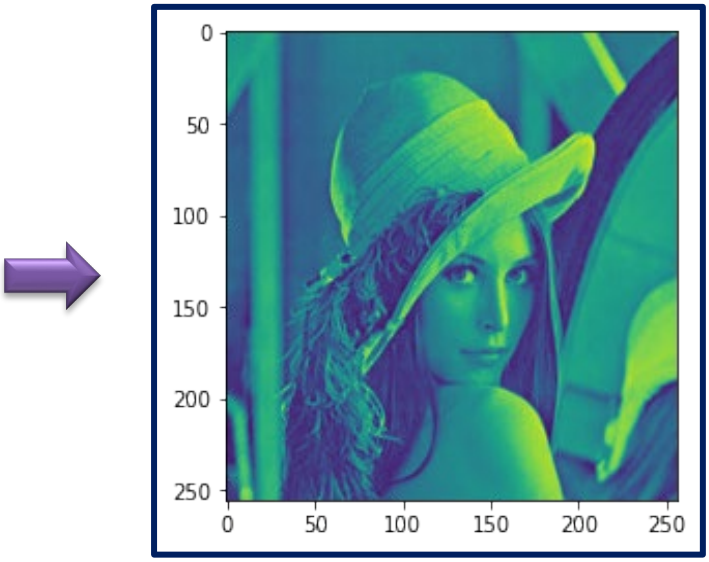
```
[17, 18, 24, 47, 99, 99, 99, 99]
[18, 21, 26, 66, 99, 99, 99, 99]
[24, 26, 56, 99, 99, 99, 99, 99]
[47, 66, 99, 99, 99, 99, 99, 99]
[99, 99, 99, 99, 99, 99, 99, 99]
[99, 99, 99, 99, 99, 99, 99, 99]
[99, 99, 99, 99, 99, 99, 99, 99]
[99, 99, 99, 99, 99, 99, 99, 99]
```

Result:

Input Image ***Output Image***







Code :

https://colab.research.google.com/drive/1UNtsv9Xewu_nQQsQqLbPin8TDKEw9Va3?usp=sharing

Discussion:

Colour – to YCbCr :In this phase, we convert R, G, B to Y, Cb, Cr model. Here Y is for brightness, Cb is color blueness and Cr stands for Color redness. We transform it into chromium colors as these are less sensitive to human eyes thus can be removed.

Compression: We apply Direct cosine transform on each block. The discrete cosine transform (DCT) represents an image as a sum of sinusoids of varying magnitudes and frequencies. The DCT coefficient values can thus be regarded as the relative amount of the 2D spatial frequencies contained in the 64-point input signal. FDCT processing step lays the foundation for achieving data compression by concentrating most of the signal in the lower spatial frequencies.

Quantization: In the Quantization process, we quantize our data using the quantization table. After output from the FDCT, each of the 64 DCT coefficients is uniformly quantized in conjunction with a 64-element Quantization Table, which must be specified by the application (or user) as an input to the encoder. Each element can be any integer value from 1 to 255, which specifies the step size of the quantizer for its corresponding DCT coefficient. The purpose of quantization is to achieve further compression by representing DCT coefficients with no greater precision than is necessary to achieve the desired image quality.

Zig-zag scanning: In serialization, we perform the zig-zag scanning pattern to exploit redundancy.

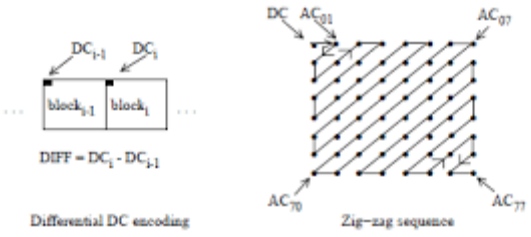


Figure 3. Preparation of Quantized Coefficients for Entropy Coding

Separately encode the DC element:We apply DPCM (differential pulse code modeling) on DC elements. DC elements are used to define the strength of colors. After quantization, the DC coefficient is treated separately from the 63 AC coefficients. The DC coefficient is a measure of the average value of the 64 image samples. Because there is usually strong correlation between the DC coefficients of adjacent 8x8 blocks, the quantized DC coefficient is encoded as the difference from the DC term of the previous block in the encoding order (defined in the following), as shown in Figure 3.

Entropy encoding: In the last stage, we apply to encode either run-length encoding or Huffman encoding. The main aim is to convert the image into text and by applying any encoding we convert it into binary form (0, 1) to compress the data. The final DCT-based encoder processing step is entropy coding. This step achieves additional compression losslessly by encoding the quantized DCT coefficients more compactly based on their statistical characteristics. The JPEG proposal specifies two entropy coding methods - Huffman coding [8] and arithmetic coding 1151. The Baseline sequential codec uses Huffman codingIt is useful to consider entropy coding as a 2-step process. The first step converts the zig-zag sequence of quantized coefficients into an intermediate sequence of symbols. The second step converts the symbols to a data stream in which the symbols no longer have externally identifiable boundaries. The form and definition of the intermediate symbols is dependent on both the DCT-based mode of operation and the entropy coding method. Huffman coding requires that one or more sets of Huffman code tables be specified by the application. The same tables used to compress an image are needed to decompress it. Huffman tables may be predefined and used within an application as defaults, or computed specifically for a given image in an initial statistics-gathering pass prior to compression. Such choices are the business of the applications which use JPEG;

the JPEG proposal specifies no required Huffman tables.

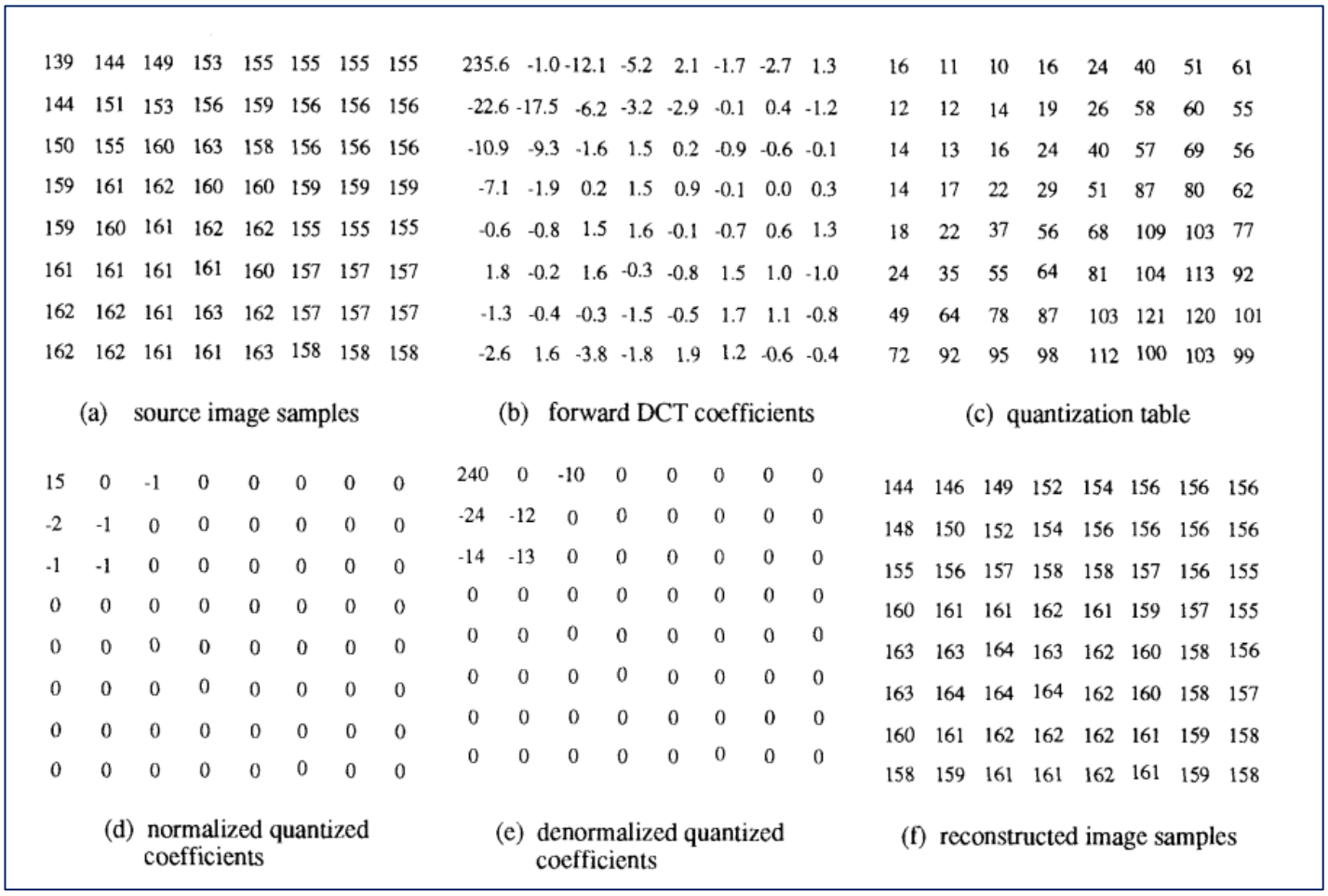


Fig. Depiction of image compression and decompression.