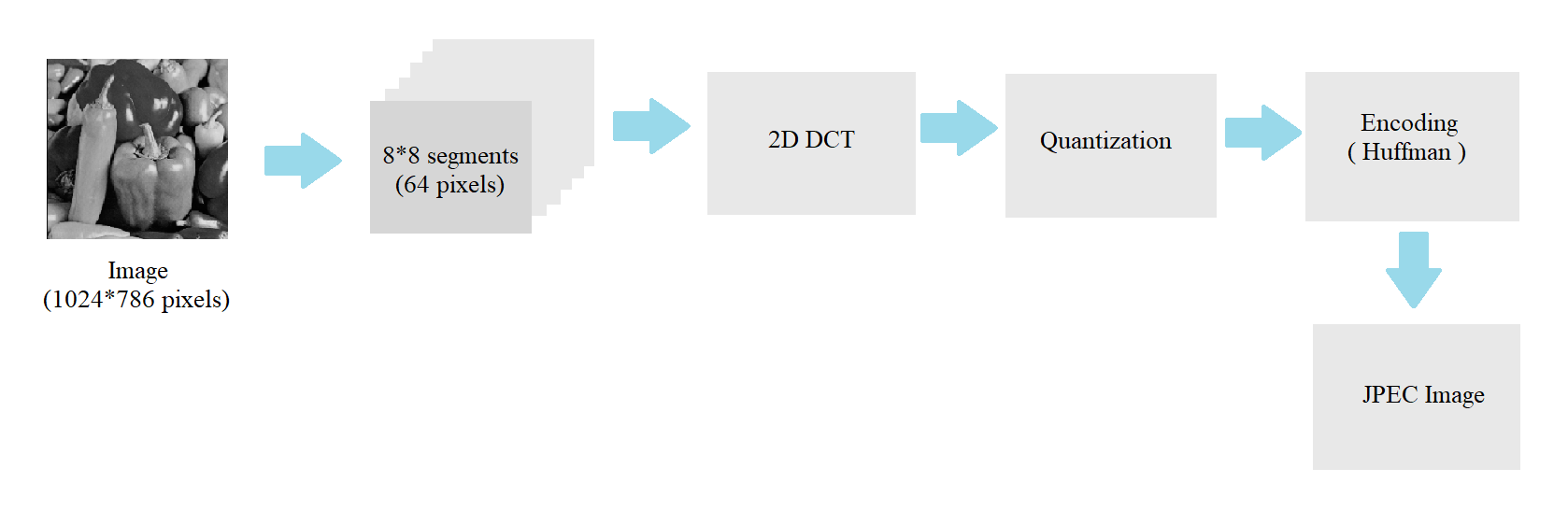
**NDA ENCODER**

NDA encoder is used to compress data using different algorithmic techniques, The proposed architecture is given below, which stores the image data in a compressed format.

An image is divided into subsegments of 64-pixel size. Each segment undergoes a transformation using the Discrete Cosine Transform (DCT) and produces a series of coefficients that represent the image's frequency content. These coefficients are organized into a matrix called a quantization table.



The quantization table contains entries used to quantize the frequency coefficients to reduce the amount of data required to store the image. Quantization involves dividing each coefficient by a corresponding value in the quantization table and rounding the result to the nearest integer.

After quantization, the compressed data is further compressed using variable-length coding. This involves assigning shorter codes to more frequently occurring values and longer codes to less frequently occurring values, resulting in a reduction in the amount of data required to represent the compressed image.

When a JPEG image is decompressed, the compressed data is first decoded using the variable-length coding, and then the quantized frequency coefficients are reconstructed using the quantization table. The inverse Discrete Cosine Transform (IDCT) is then applied to the reconstructed coefficients to obtain the image data in the spatial domain, which can be displayed on a screen or printed on paper.

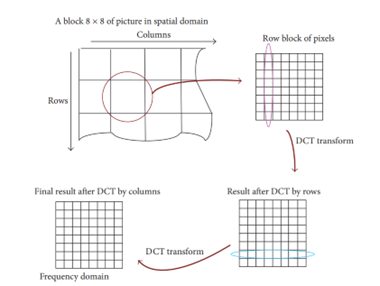
**2D DCT**

A 2D Discrete Cosine Transform (DCT) can be computed by applying a 1D DCT to each row of the image, followed by a 1D DCT on each column of the resulting matrix of transformed rows.

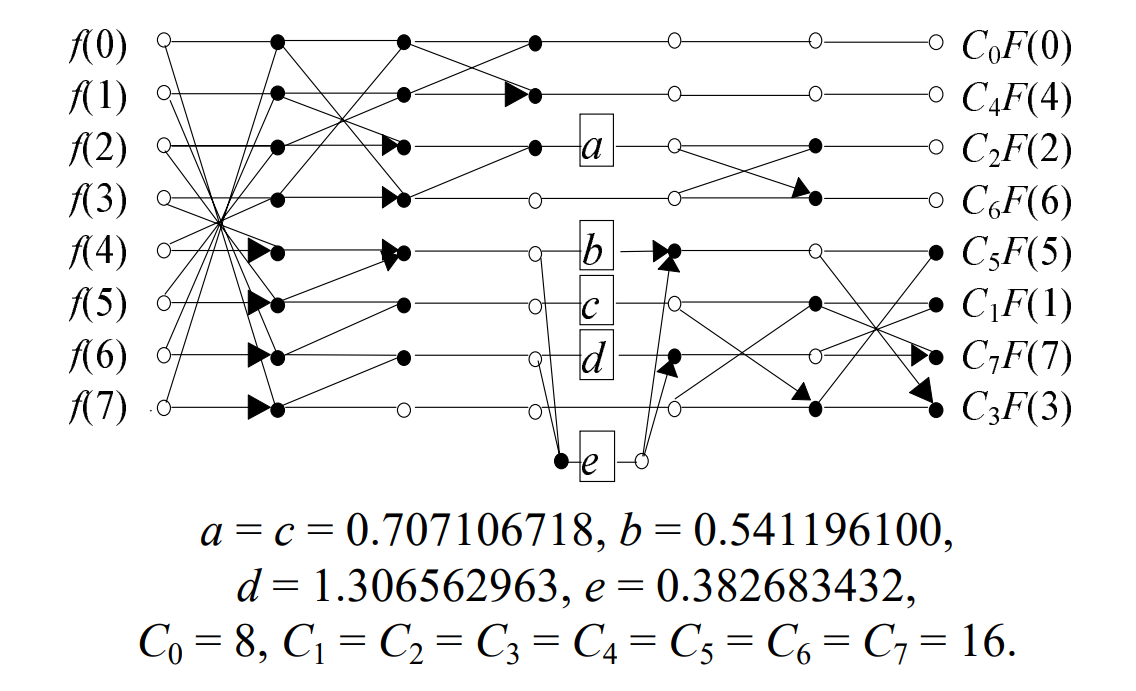
Here's how the process works:

* Apply a 1D DCT to each row of the image:
  + For each row of the image, apply a 1D DCT to the row's pixel values.
  + The result will be a matrix of transformed row vectors.
* Apply a 1D DCT to each column of the matrix:
  + For each column of the matrix, apply a 1D DCT to the column's transformed row vector.
  + The result will be a matrix of transformed coefficient values.

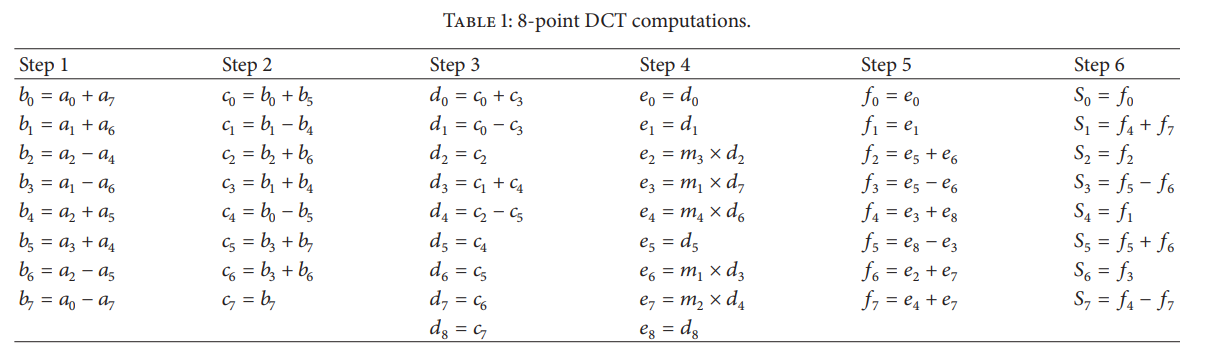
The resulting matrix of transformed coefficients represents the image in the frequency domain. The lower frequency coefficients are located near the upper left corner of the matrix, while the higher frequency coefficients are located near the lower right corner.



**1D DCT algorithm**

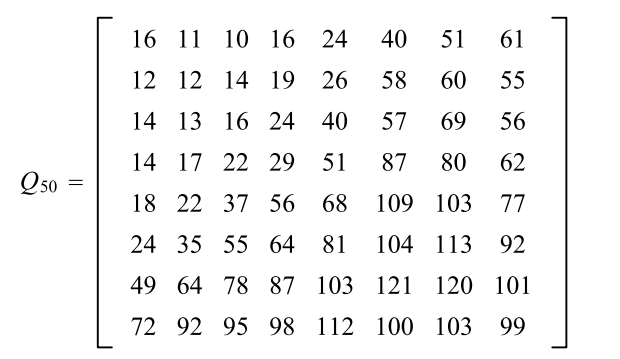
****

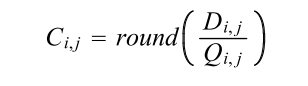
The proposed algorithm is using only 29 addition and 5 multiplications, which is comparatively very less hardware usage by the primitive methods



**Quantization**

Quantization is achieved by dividing each element in the transformed image matrix D by the corresponding element in the quantization matrix and then rounding to the nearest integer value. For the following step, the quantization matrix Q50 is used.





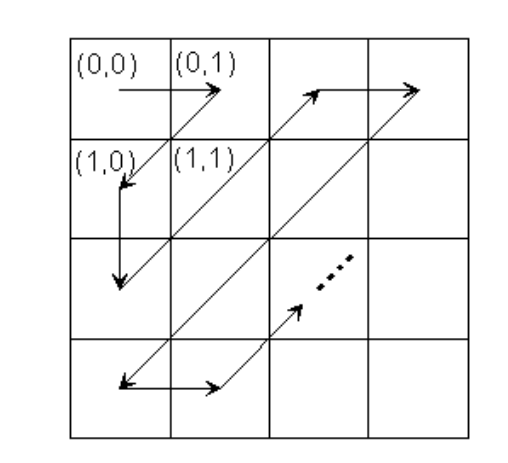
Where matrix D is the output after 2D DCT transformation.

**Huffman coding**

Huffman coding is a variable-length entropy coding technique used to compress data by assigning shorter codes to more frequently occurring symbols in the data. It can be used as a post-processing step after 2D DCT (Discrete Cosine Transform) to further reduce the amount of data needed to represent an image.

Overall, the advantages of using Huffman coding after 2D DCT include efficient use of memory, high compression ratios, a flexible coding scheme, and fast decoding.

Before storage all coefficients of Matrix C are converted by an encoder to a stream of binary data (01101011….) by the order given in the below figure.



This involves assigning shorter codes to more frequently occurring values and longer codes to less frequently occurring values, resulting in a reduction in the amount of data required to represent the compressed image.

**Reference**

1. S. E. Tsai, S. M. Yang, "A Fast DCT Algorithm for Watermarking in Digital Signal Processor", Mathematical Problems in Engineering, vol. 2017, Article ID 7401845, 7 pages, 2017. https://doi.org/10.1155/2017/7401845
2. M. Kovac and N. Ranganathan, “VLSI circuit structure for implementing JPEG image compression standard,” US patent US5659362, 1997.
3. M. Kovac and N. Ranganathan, “JAGUAR: a fully pipelined VLSI architecture for JPEG image compression standard,” Proceedings of the IEEE, vol. 83, no. 2, pp. 247–258, 1995.
4. K. Cabeen and P. Gent, “Image Compression and Discrete Cosine Transform,” College of Redwoods. <http://online.redwoods.cc.ca.us/instruct/darnold/LAPROJ/Fall98/PKen/dct.pdf>
5. https://www.youtube.com/watch?v=Q2aEzeMDHMA&t=15s