Image compression using 2D Discrete Cosine Transform (DCT)

Dr. C. Jeyalakshmi[1,], drcjeyalakshmiece@krce.ac.in
Faculty, Department of ECE, K. Ramakrishnan college of engineering, Tamilnadu
Zulaiha Rosni R[2], Saraal S[3], Subishna R[4], Priyadharshini E[5]
zulaiharosnir.ece24@krce.ac.in[2], saraalsathiyanathan@gmail.com[3], subishnar3@gmail.com[4], priyaelangovan.2005@gmail.com[5]
Students, Department of ECE, K. Ramakrishnan college of engineering, Tamilnadu

Abstract— Image compression aims to reduce the storage and transmission requirements of digital images. This work is implemented by block-based image compression which used 2D Discrete Cosine Transform and this technique is widely used in JPEG. The input is grayscale image. The grayscale image is divided into 8x8 blocks. Each block is converted from spatial domain to frequency domain using Discrete Cosine Transform. some of the high frequency information which are less significant are removed from the image and low frequency information which are more significant are kept to make perfect compression of the given signal. Then Inverse Discrete Cosine Transform is applied to the low frequency data to rebuild the image from compressed frequency data. The quality of compressed image is computed using Peak Signal-to-Noise Ratio (PSNR). The result shows that better compression can be made with minimal loss in optical quality and high spot the success of 2D Discrete Cosine Transform in applications of image compression.

Keywords— Grayscale Image, Image division, Discrete Cosine Transform (DCT), low frequency DCT Coefficients, Inverse Discrete Cosine Transform (IDCT), Peak Signal-to-Noise Ratio (PSNR).

I. INTRODUCTION

Digital images consume lot of space and for storage space and needs notable bandwidth for transmission. Image compression offers the reduction in storage space and notable bandwidth without compromising the quality of the image. By reducing the storage space of image, it can be loaded faster on websites. This will also reduce time required for downloading time of the image. With the boom of digital media, image data handling is essential. Size of files depend on pixels that an image contains.

For High-resolution images, file size will also be high as it contains millions of pixels. The large size of files leads to increased storage space and increases the time required for transmission, uploading and downloading. Image compression label these challenges by reducing file size and transmission time and sustaining acceptable optical quality. The major aim of image compression is to save storage space and bandwidth. When the image size is smaller websites can be loading faster and also the server load can be reduced.

II. LITERATURE REVIEW

Efficient Learned Image Compression with Unevenly Grouped Space-Channel Contextual Adaptive Coding is surveyed by Dailan et al [1]. This work surveyed a learned image compression model which *used* uneven channel- conditional adaptive coding—grouping channels unevenly based on energy compaction. This helps to improve coding efficiency without reducing speed. It bears fast decoding. This results in attaining state-of-the-art trade-offs between compression performance and runtime**.** Enhanced Residual SwinV2 Transformer for Learned Image Compression was studied by Wang et al.[2]. This paper initiates the structure based on Residual SwinV2 transformer. It incorporates a feature enhancement module (several convolutional layers) followed by transformer attention in coding steps that helps to capture image structure more productively.

Yaojun Wu et al [3] surveyed Learned Block-based Hybrid Image Compression. In order to overcome the memory footprint and slow decoding due to autoregressive entropy models They propose block partitioning (compressing image in blocks) plus an explicit intra prediction module (to model correlation between blocks), and a boundary-aware post-processing module to reduce block artifacts. This paper reduces the decoding time substantially.

Jiang et al [4], surveyed An End-to-End Compression Framework Based on Convolutional Neural Networks. This article combines two CNN'S: The purpose of first CNN is to learn a compact representation (ComCNN), which compresses to a representation that is then encoded using a standard codec (e.g. JPEG, BPG, JPEG2000). The purpose of second CNN is to rebuilt the decoded image. (RecCNN). The reconstruction is to recuperate the quality of image.
Jamil et al surveyed [5] Learning-Driven Lossy Image Compression. This survey deals with lossy image compression techniques which uses machine learning that includes autoencoders (standard, variational), CNNs with hyper-priors, RNNs, GANs, PCA, and fuzzy clustering. The aim of this paper is to suggest a guide to select appropriate MI-based compression methods depending on application constraints. The paper classifies the approaches, discusses trade-offs between bitrate, distortion, computational complexity, and inspects how redundant or peripheral information is removed.

Uddehal et al [6] surveyed Image Segmentation for Improved Lossless Screen Content Compression. This paper deals with the compression of screen content images which combines text, graphics and natural content. In this paper Segmenting SCIs is preferred to separate synthetic (e.g. text, graphics) versus natural regions. Then different probability models and entropy coding are applied for each. Finally, they achieve up to ~11.6% rate reduction over HEVC (High Efficiency Video Coding) and ~1.52% over previous Soft Context Formation methods. Pansare and Jadhav compared multiple classical and hybrid compression approaches[7] to analyze trade-offs in compression ratio, computational cost, and quality (lossless vs lossy). Their aim is to find which methods work well in which contexts (depending on allowed distortion, image type).

Bourai et al [8] Advanced Image Compression Techniques for Medical Applications. This paper covers both traditional methods (JPEG, JPEG2000, PNG) as well as deep learning approaches (CNN, RNN, LSTM) to focus on compression on medical imaging. This survey also discusses on compression defects such as block artifacts and ringing. The proposed technique highlights the crucial challenges namely balancing compression versus diagnostic quality of image and the requirement for better evaluation metrics and standardized datasets. Saudagar[9] surveyed Biomedical Image Compression Techniques for Clinical Image Processing**.** Here author explores three main AI-based approaches for compressing medical/clinical images. Neural networks, fuzzy logic & neuro-fuzzy logic, and relational coding are used for inter-band coefficients. This shows improved compression under resource constraints. Image Compression Techniques Review was studied by Singh and Garg[10]. This paper is a review of classical algorithmic image compression methods namely Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Vector Quantization (VQ), and Fractal Compression. This paper discusses their principal, advantages and disadvantages such as compression ratio vs distortion, computation cost mostly for grayscale images.

Content-Oriented Learned Image Compression was studied by Meng Li et al[11] . Human eyes have different sensitivities to different content, so the image content also needs to be considered. Most learning-based image compression methods are unlabeled and do not consider image semantics or content when optimizing the model however, image content also needs to be considered. A content-oriented ima*-ge compression method, which handles different kinds of image contents with different strategies are proposed in this paper. Extensive experiments show that the initiated method achieves competitive subjective results compared with state-of-the-art end-to-end learned image compression methods or superior methods.

Ze Cui studied Asymmetric Gained Deep Image Compression with Continuous Rate Adaptation [12]. With the development of deep learning techniques, the combination of deep learning with image compression has gain lots of recognition. In this paper a continuously rate adjustable learned image compression framework, Asymmetric Gained Variational Autoencoder (AG-VAE). AG-VAE make use of a pair of gain units to reach discrete rate adaptation in one single model with a insignificant additional evaluation. By using exponential interpolation, continuous rate adaptation is achieved without compromising performance. And by using exponential interpolation, without compromising performance continuous rate adaption is achieved.

Semantics-to-Signal Scalable Image Compression with Learned Revertible Representations was studied by Kang Liu[13].

In this paper, scalable image compression method that connects semantic information with signal representations are put forward. It uses learned reversible (invertible) transformations to encode images in a way that can be scaled — i.e., parts of the representation depending on desired fidelity or bitrate can be truncated or refined. The goal of this method is to stabilse optimal quality, semantic fidelity, and compression efficiency. In the final section of this survey, conclusions regarding its performance which are well compared to other scalable and non-scalable compression schemes are explored.

Ekta Soni et al surveyed on MRI Image Compression Using Asymmetric Wavelet Analysis [14]. It demonstrates the compression of MRI Image Compression using Asymmetric Wavelet Analysis. While compressing the proposed technique point up preferable preservation of diagnostically applicable. In this paper , asymmetric wavelet transform is used for the improvement of compression performance for MRI data, by taking both fidelity and perceptual quality in consideration.

The authors propose a scheme where traditional distortion metrics and adversarial loss is used to better preserve perceptual quality. Deep networks are employed as encoder / decoder, with convolutional architectures trained adversarially therefore the compressed and rebuilt images are more optically pleasing, reducing artifacts common in standard compression methods. This paper shows improved perceptual performance over conventional methods under similar bitrate constraints.

III. METHODOLOGY

A. *Discrete Cosine Transform*

Discrete Cosine Transform is a mathematical technique which is used to convert spatial domain data (like pixels in an image) into frequency domain data. It takes a signal (i.e., Image) and return it in term of sum of cosine waves of different frequencies. It is as same as Fourier Transform. The only difference is that Discrete Cosine Transform uses only Cosine Transform which is better for signals like image. An image is made up of pixels and the pixels will have some intensity values. In an image Low frequency refers to the region which takes more time to change and High frequency refers to the region which takes less time to change (eg. edges in an image). Block Diagram of Image Compression using DCT is shown in fig.1.

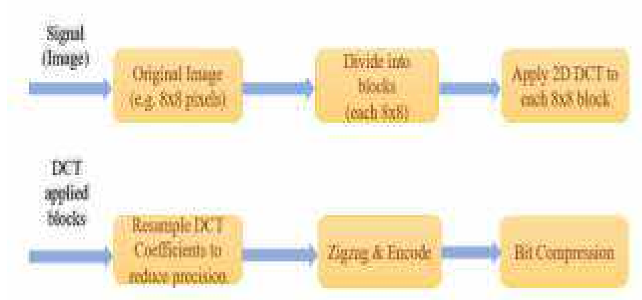

Figure: 1. Block Diagram of Image Compression

DCT converts image blocks which are 8x8 pixels into a set of coefficients, which will give the information about how much low and high frequency information are present in it. Most of the images contain key information in low frequencies and high frequencies are not much noticeable to the human eye. So, some of the high frequency information are removed from the image and low frequency information are kept to make perfect compression of the given signal (i.e.,image). After applying DCT to all blocks , only the top-left 4×4 DCT coefficients (low frequencies) are kept using masking.For an image block f(x,y) of size NxN, the 2D DCT is:

$$F(u,v)= \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y).\cos[\frac{\pi}{N}(x+\frac{1}{2})u].\cos[\frac{\pi}{N}(y+\frac{1}{2})v]$$

f(x,y):pixel intensity at(x,y)

F(u,v):DCT coefficient at (u,v)

$\alpha(u), \alpha(v)$: *same normalization as before.*

B. *Inverse Discrete Cosine Transform*

Discrete Cosine Transform is the reverse operation of Discrete Cosine Transform. Discrete Cosine Transform converts spatial domain data (like pixels in an image) into frequency domain data, Inverse Discrete Cosine Transform again converts the frequency domain to spatial domain. Block Diagram of applying Inverse Discrete Cosine Transform is shown in fig.2. Inverse Discrete Cosine Transform is taken for the Discrete Cosine Transform coefficients and reconstruct the original signal(image).The sum of cosine functions are used to take Inverse Discrete Cosine Transform.

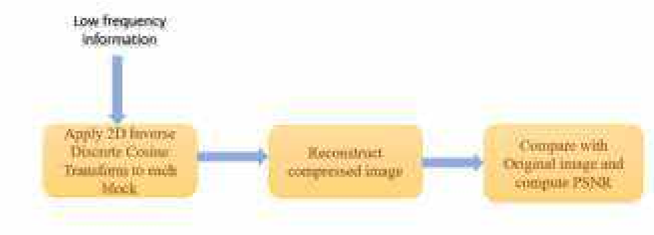

Figure: 1. Block Diagram of applying Inverse Discrete Cosine Transform

The sum of cosine functions are used to take Inverse Discrete Cosine Transform.

Formula of Inverse Discrete Cosine Transform is given by,

$$x_n=\sum_{k=0}^{N-1} \alpha(k)X_k \,.\cos[\pi /N(n+\tfrac{1}{2})k]$$

C. PSNR Computation

PSNR stands for Peak Signal-to-Noise Ratio. It is used to measure the quality of compressed image compared to original image. It is defined as the ratio of maximum possible signal value and distortion. If the value of PSNR value is high it implies that the quality of compressed image is better (i.e., quality is closer to the original). Low PSNR value implies that the quality of compressed image is poor (i.e., compressed image contains more distortion). PSNR value is measured in decibels(db).

Mean Squared Error (MSE) is calculated by taking average squared difference between pixels of original image and compressed image. This MSE is used to quantify the error initiated by compression.The formula of PSNR is given by,

$$\text{PSNR}=20.\log_{10}(\frac{MAX_I}{\sqrt{MSE}})$$

If the difference between original and compressed image are equal , then the value of MSE=0. Since log(0) is undefined the PSNR value is taken as infinite in this case. PSNR values and corresponding quality are shown in table1.

From table1 the Peak signal-to-Noise Ratio and quality of corresponding compressed image are known. If it is greater that 50 decibals the quality is Excellent and virtually there is no perceptible difference from the original grayscale image. If it range from 40-50 decibals the quality is very good and the difference is barely noticeable to the human eye. When the PSNR value is in the range of 30-40 decibals the quality is acceptable and there is a Slight degradation. 25-30 decibal implies that the quality of the image is fair and there is a noticable compression artifacts or blurring. 20-25 decibals says that the quality of the image is poor and there is a significant distortion, often the image is not suitable for use. If the PSNR value is lesser than 20 decibals then the quality of the image is very poor and there will be major artifacts.

Table: 1. PSNR Values and its quality

PSNR (dB)	Quality
> 40	Excellent (almost lossless)
30 – 40	Good (visually acceptable)
20 – 30	Poor (visible artifacts)
< 20	Very poor

IV.RESULTS AND DISCUSSION

A. Input and application Discrete Cosine Transform on it

A grayscale image is given as input. A grayscale image is an image which contains only intensity information in each pixel and contains only the shades of gray, it ranges from black to white. Black shade has minimum intensity and white shade has maximum intensity. There is no color information in a grayscale image and it has only one channel.

In an 8-bit grayscale image, each pixel is represented by a single value from 0 to 255. The value 0 indicates black shade (maximum intensity) and the value of 255 indicates white shade (minimum intensity). The values between 0 and 255 represents varying shades of gray. A grayscale image is shown in fig.3.

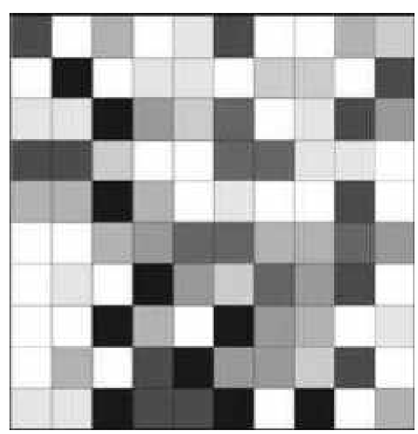

Figure:3. Grayscale image 0-255 pixel

Compression is performed to decrease the size of the data (e.g. grayscale image). A reknown method for compression of the image is frequency-based compression**.** For this, should be in frequency domain (spatial domain should be converted into frequency domain).To convert spatial domain into frequency domain Discrete Cosine Transform is used. DCT helps in representing image data in terms of frequency components. The DCT cannot be applied on the entire image. So, the image is divided into small square blocks (i.e. 8x8 pixel).

These blocks are processed individually to keep the calculation achievable. The division into blocks also aligns with human perception, which is locally sensitive. Each 8×8 block contains 64-pixel intensity values. These values vary smoothly across space, forming image patterns. The purpose of applying DCT is to analyse how these intensity values vary. Discrete Cosine Transform transforms the block (from pixel values) into a combination of cosine functions. These cosine functions represent different spatial frequencies**.** Low frequencies imply that the block takes more time to change.

High frequencies (edges, fine details, or noise) implies that the block can change quickly. After applying DCT to each block, the block becomes a set of DCT coefficients. The top-left coefficient is the DC component and it represents the average value. Other coefficients are AC components and they represents the frequency variations. Most important optical information tends to be concentrated in low-frequency coefficients. The high-frequency coefficients often have smaller values and can be removed. The process of image compression till applying DCT is shown in fig.5

This concentration of energy is a key factor which makes DCT useful for performing compression. The transformation is reversible — original data can be approximated through Inverse DCT**.** But before that, data is compressed by discarding less important coefficients. So, the DCT is the key step that prepares the image for efficient compression. Each block's DCT output is used in the consecutive stages of compression (quantization, encoding, etc.). Up to this stage, the image has been analyzed and transformed — but not yet compressed.

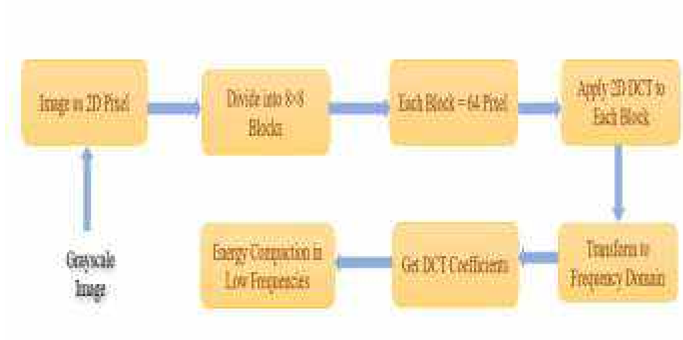

Figure:5. Block diagram for Discrete Cosine Transform

B. Inverse Discrete Cosine Transform

After applying DCT, each 8×8 image block is converted to the frequency domain. This portrayal contains both low and high-frequency coefficients. To compress the image, many high-frequency coefficients are eliminated or zeroed out. This reduces data (image) size but initiates some loss in data. The remaining coefficients are kept for rebuilding. These retained coefficients catch the most important optimal information.

Now, to rebuild the image, Inverse DCT (IDCT) is applied to each block. IDCT transforms the frequency-domain data to the spatial domain. It reconstructs pixel intensity values from the cosine components. The reconstructed block is only an approximation of the original block. If all DCT coefficients were retained, the reconstruction would be exact. The approximation is slightly different because of the lossy compression.

If more coefficients are retained the compressed image will be closer to the original. If less coefficients are retained then the compressed image will have more loss in given data. IDCT is mathematically the inverse of DCT and uses the same cosine basis. It combines the weighted cosine functions (via the coefficients) to restore pixel values. Then the restored 8x8 blocks are rearranged into the full image. By this way the final compressed and rebuilt image is formed. The optical quality of the output depends on how many coefficients were preserved. Despite some data loss, the human eye is often unable to notice minor differences. This is because human eyes are more sensitive to low-frequencies (shapes, structure).

IDCT takes advantage of this by reconstructing key features using only low-frequency data. The resulting image will retain its general appearance and structure. Datas like Fine details, textures, or sharp edges may be slightly blurred. This trade-off between compression ratio and quality is typical of lossy compression. After applying IDCT, we can evaluate the quality of the image using Peak Signal-to-Noise Ratio (PSNR). IDCT completes of image compression and reconstruction. This DCT-IDCT framework is foundational to standards like JPEG. Fig.6 shows the block diagram of IDCT.

Figure:6. Block Diagram of Inverse Discrete Cosine Transform

C. *Original and Compressed Image*

Original Image is shown in fig.7,

Figure:7. Original Image

Compressed Image is shown in fig.8,

Figure:8. Compressed Image

PSNR: 54.70dB

V.CONCLUSION

Image compression is necessary for efficient storage, transmission, and processing. 2D Discrete Cosine Transform (DCT) is a widely used technique in lossy image compression (which removes unessential data). Discrete Cosine Transform transforms spatial domain of a data into frequency domain. The 2D DCT is applied to each block (8×8 pixel) of a grayscale image. This transformation divides the image into low and high-frequency components. Low-frequency region carries important optical information. High-frequency components often contain unwanted data that can be discarded. By retaining only low-frequency coefficients, the data is reduced. This process achieves compression by eliminating insignificant data.

Inverse Discrete Cosine Transform (IDCT) is then used to rebuild the image from the retained coefficients. Although the reconstructed image is not identical to the original, the visual quality will remain as an acceptable one. The level of compression can be adjusted by varying the number of coefficients retained. Higher compression leads to loss of more data and hence reduces file size. Lower compression retains more detail and hence increases the file size. The performance of the compression is computed by Peak Signal-to-Noise Ratio (PSNR). A high PSNR indicates good reconstruction quality, with minimum distortion. The human eye is less sensitive to high-frequency losses so, Discrete Cosine Transform based compression is effective. This method takes advantage of human visual perception to achieve efficiency. It forms the basis of widely used standards such as JPEG.

Therefore, image compression using 2D DCT is both theoretically sound and practically valuable. In the future, DCT may combine with wavelet transforms, neural networks, or other mathematical models to improve compression ratios and optical quality.

V.REFERENCES

[1] He, Z. Yang, W. Peng, R. Ma, H. Qin, and Y. Wang, "ELIC: Efficient Learned Image Compression with Unevenly Grouped Space-Channel Contextual Adaptive Coding,"arXiv preprint arXiv:2204.12068, 2022.

[2] G. Wang, X. Zhang, J. Yu, and Z. Ma**,** "Enhanced Residual SwinV2 Transformer for Learned Image Compression,"*arXivpreprint arXiv:2311.06708*, 2023.

[3] Y. Wu, Z. Liang, T. Dumas, F. Yang, W. Liu, and P. Frossard,**"Learned Block-Based Hybrid Image Compression, "**IEEE Transactions on Image Processing, vol. 31, pp. 3647–3660, 2022.

[4] Q. Jiang, P. Zhang, D. Wu, Z. Guo, and X. Zhang, "An End-to-End Compression Framework Based on Convolutional Neural

Networks, "IEEE Transactions on Multimedia, vol. 21, no. 6, pp. 1417–1430, 2019.

[5] S. Jamil, F. Ahmed, and M. Rizwan, "Learning-Driven Lossy Image Compression: A Comprehensive Survey, "Multimedia Tools and Applications, vol. 82, no. 8, pp. 11725–11758, 2023.

[6] P.Uddehal and M. Ghanbari,"Image Segmentation for Improved Lossless Screen Content Compression," IEEE Access, vol. 10, pp. 8453–8467, 2022.

[7] A. S. Pansare and D. V. Jadhav, "A Comparative Study of Classical and Hybrid Image Compression Techniques," International Journal of Computer Applications, vol. 182, no. 48, pp. 1–7, 2019.

[8] B. Bourai, R. Dey, and S. Saba, "Advanced Image Compression Techniques for Medical Applications,"Biomedical Engineering Letters, vol. 13, no. 2, pp. 123–145, 2023.

[9] R. Saudagar,"Biomedical Image Compression Techniques for Clinical Image Processing," Biomedical Signal Processing and Control, vol. 79, no. 105020, 2022.

[10] A. Singh and S. Garg, "Image Compression Techniques Review, "International Journal of Computer Science and Information Technologies, vol. 6, no. 1, pp. 29–34, 2018

[11] M. Li, Y. Liu, and Z. Xiong,"Content-Oriented Learned Image Compression,"arXiv preprint arXiv:2201.08912, 2022.

[12] Z. Cui, J. Zhang, and W. Lin,"Asymmetric Gained Deep Image Compression with Continuous Rate Adaptation,"arXiv preprint arXiv:2303.05211, 2023.

[13] K. Liu, X. Chen, and H. Li,"Semantics-to-Signal Scalable Image Compression with Learned Revertible Representations, "IEEE Transactions on Image Processing, vol. 32, pp. 2911–2924, 2023.E. Soni and K. Patel,"MRI Image Compression Using Asymmetric Wavelet Analysis," International Journal of Biomedical Imaging, vol. 2021, Article ID 5598743, 2021.