Journal of Integrated
SCIENCE & TECHNOLOGY

Optimal machine learning model based medical image compression techniques for smart healthcare

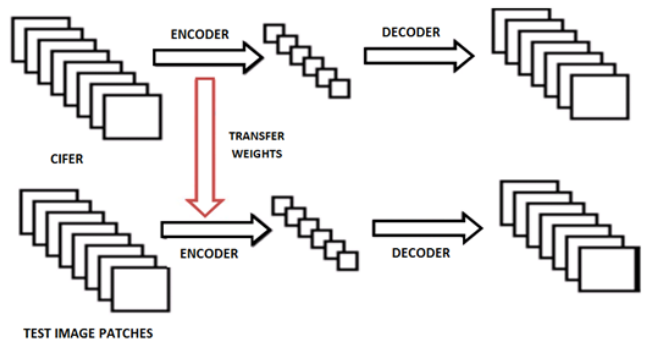
Bharath K.N.[1*], K. Suresh Babu[2]

[1]*Department of ECE, Dayananda Sagar Academy of Technology and Management (DSATM), Bangalore, India.* [2]*Department of ECE, University of Visvesvaraya College of Engineering (UVCE), Bangalore, India.*

Received on: 11-Dec-2023, Accepted and Published on: 27-Mar-2024

Article

ABSTRACT



In healthcare systems, minimizing storage needs, transmission bandwidth, and processing expenses requires effective medical picture compression. This paper provides a thorough analysis of the many methods and strategies used in medical image compression. The difficulties of precisely compressing medical picture data are examined, particularly the requirement to preserve diagnostic quality at high compression ratios. More modern strategies like deep learning-based compression techniques are contrasted with more established ones like JPEG and JPEG2000 compression. The usage of neural networks, autoencoders, and generative adversarial networks (GANs) as well as other lossless and lossy compression techniques are also explored in this research. The suggested method makes use of CNN-RNN-AE to learn a condensed version of the original image, which had structural information. Multilayer perceptron's (MLPs) were utilized for lossless image compression, while autoencoders and generative adversarial networks (GANs) were employed for lossy compression. The original image was then recovered by decoding the encoded image using a high-quality reconstruction approach. The optimal compression technique that has been provided fits in with the current image codec standards. A variety of experiment outcomes were compared with JPEG, JPEG2000, binary tree, and optimal truncation in terms of space saving (SS), reconstructed image quality, and compression efficiency. The results support the effectiveness of the designed strategy.

Keywords: Machine learning, Image compression, Image reconstruction, Autoencoders

INTRODUCTION

In data storage and transit, image compression is crucial, particularly in light of the data explosion that is happening at a rate much faster than Moore's Law.[1] Finding and recovering the pixels is a difficult undertaking because of the extremely intricate unknown relationships between them. In order to successfully recover a picture from a well-compressed representation, either losslessly or with some loss, we want to build and test networks that can do so. 5G connects various items in the world and allows them to communicate with one another through the Internet with little to no human intervention.[2] Mobility management is specifically utilized to enable every mobile node to connect to the internet independent of location or movement. Increased expectations for creative applications that break through and outpace current constraints are a result of the 5G era. Additionally, in order for smart home networks to allow anytime, anywhere remote access to applications and sensors (home IoT), mobility management is required. The DMM is used by 5G networks' smart homes to enable mobile nodes, and remote access also manages the related nodes. Critical threats that can be readily managed and access smart home gadgets are the primary problems with remote access. Therefore, it is crucial to protect and transfer user data via an optimum way; as a result, safe routing is more crucial in smart homes. By automating machinery, businesses are able to use cutting-edge software platforms for monitoring, controlling, and making subsequent production process evolution more stunning. Image sensors make up the majority of sensor networks used in manufacturing automation. Picture sensors are essential in security and safety applications because they can recognize and transmit the data that makes up a picture.

*Corresponding Author: Bharath K N, Department of ECE, DSATM
Email: bharath-ece@dsatm.edu.in

Cite as: *J. Integr. Sci. Technol., 2024, 12(5), 821.*
URN:NBN:sciencein.jist.2024.v12.821

The difficulties involved in medical diagnosis and therapy highlight how urgently improvements in medical image compression are needed.[2] For patients to receive appropriate care in the current healthcare environment, fast and reliable diagnostic information must be available. Nonetheless, storage capabilities, unequal access to cutting-edge imaging technologies, and restricted bandwidth for data transmission are common challenges faced by healthcare facilities. In contexts with little resources and distant locations with little medical infrastructure, these difficulties are especially noticeable. Utilizing picture compression methods, the study seeks to lessen these challenges. In order to maintain diagnostic integrity while maximizing the storage and transfer of enormous volumes of medical imaging data, image compression is an essential technique. The research aims to improve patient outcomes by facilitating seamless medical data sharing across healthcare networks, improving accessibility to diagnostic resources, and developing efficient compression algorithms that are tailored to the unique requirements of medical imaging. As a result, picture compression is shown to be a key component in resolving the complex issues associated with medical diagnosis and treatment, providing a route towards more effective and fair healthcare delivery.[3]

Figure 1 shows the IoT application used for 5G which includes Internet of Vehicles (IoV), smart health care, smart home and smart city. The central processing system receives several images from each image sensor at any given time. Industrial automation equipment frequently comes with sophisticated, excellent cameras that can record images at the required resolution. Images can be supplied in low quality so that proper processing cannot be done on them in order to decrease the transfer rate. Utilizing a compression technique is a way to lower the size of an image without compromising its quality.

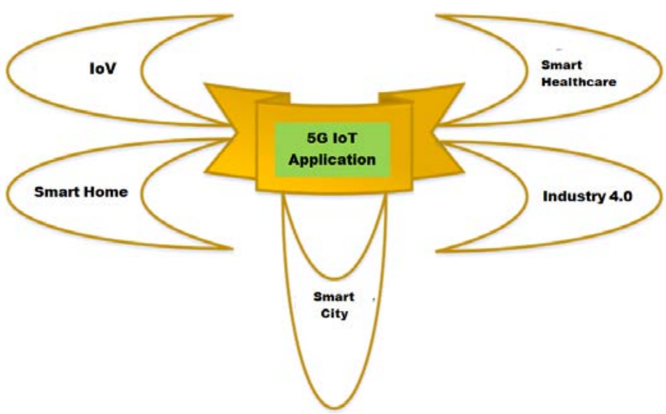


Figure 1. Security in 5G

There are unique challenges associated with incorporating AI into 5G networks. It is necessary to create efficient processes for obtaining, classifying, and analysing the enormous volumes of data that artificial intelligence has amassed. Therefore, early adopters of AI who find solutions to these challenges will become obvious leaders when 5G networks get connected. For a very long time, people have thought of satellite communication as a separate technology from mobile networking. Next-generation satellites, based on 5G architecture, will let networks control connectivity to cars, planes, and other IoT devices in remote and rural locations. Thanks to 5G, service providers will be able to offer a range of services and the satellite industry will be able to grow beyond its existing specialty.

Because 5G networks and smart home applications have different requirements and features, the suggested picture compression algorithms are especially important. It becomes critical to transmit data efficiently when using 5G networks, which offer previously unheard-of levels of speed, bandwidth, and latency. Big data transfers, particularly real-time transfers of high-resolution medical images, are made possible by 5G networks' enormous throughput capabilities. But in order to properly utilize 5G networks, bandwidth efficiency is essential, particularly in situations where network congestion or bandwidth limitations can emerge. To improve bandwidth usage and overall data transmission efficiency over 5G networks, the suggested image compression algorithms offer a solution by minimizing the size of medical image files while preserving diagnostic quality.[3]

Furthermore, effective image data handling is critical in smart home applications where medical imaging is increasingly integrated for telehealth and remote monitoring purposes. A large quantity of data is generated by smart home devices, such as wearable health trackers and home-based diagnostic tools, and this data needs to be quickly processed and transferred. Smart home devices can reduce medical picture data size without sacrificing diagnostic quality by putting the suggested image compression techniques into practice.[4] This would increase transmission efficiency and lower storage needs for the smart home ecosystem. This improves the overall efficacy of smart home healthcare solutions while also preserving resources by enabling quicker processing and analysis of medical imaging data. It also facilitates prompt healthcare actions. The suggested image compression methods, in essence, provide a mutually beneficial match for 5G networks and smart home applications by meeting their unique needs for effective data processing, storage, and transmission. This will ultimately help to advance telemedicine, remote patient monitoring, and customized healthcare delivery.

The main contributions of this paper are:

- This study's first contribution is the use of predictive coding via multilayer perceptron (MLP) for lossless compression. This is the first time that MLP has been used to compress images. Additionally, this article discusses the MLP algorithm's performance in contrast to other approaches.
- The second contribution is the use of autoencoders and GANs for lossy compression.
- The application of our strategy in the 5G is the subject of the second contribution. There hasn't been any research on 5G sensor picture compression to date. The dataset utilized in this study was supplied in a genuine smart greenhouse and is very applicable to industrial areas.
- The final but not least contribution is that our solution has been tried and developed for both RGB and grayscale photos. However, only grayscale images have been examined in the aforementioned literatures. We should use RGB images because

some image cameras are RGB and require RGB code analysis. Grayscale photos are also analyzed in order to compare the performance of the K-Means++ method to that of the other algorithms.

Literature Survey

A variety of deep learning techniques and neural networks have been applied to image compression. enumerates the majority of neural network techniques for image compression put forth prior to 1999. Nevertheless, no effective deep learning algorithms existed during that time. For picture compression, Toderici et al.[5] presented one of the most effective deep learning methods. They outperformed JPEG by a small margin thanks to their use of recurrent neural network (RNN) architecture. They tackled the issue of JPEG compression for small photos with minimal amounts of redundant data. A network that adheres to the traditional three-stage compression process—encoding, quantization, and decoding—was proposed by Santurkar et al.[6] Two distinct RNN architectures—Long Short Term Memory (LSTM) and convolutional/deconvolutional LSTMs—are used successively for encoding and decoding. This method's drawback is that it takes a lot of time because encoding and decoding must be done repeatedly, but its benefit is that the compression ratio may be gradually increased. Recently, a team used GANs to build a compression network that produced higher compression rates and more aesthetically beautiful images. Nonetheless, their recreated representations differ from the originals in that they are generally interpreted as a distinct individual or entity. According to Theis et al.[7] proposal, compressive autoencoders use an entropy coding technique in which values in images are assigned a bit representation based on how frequently they appear in the image. The derivative of the rounding function becomes undefined when the quantization (rounding to the closest integer) approach is used for encoding, as Shirsat et al.[8] described. This issue is only resolved by a smooth approximation in the backward pass. According to Johnston et al.[9], the structural similarity index metric (SSIM) and peak signal to noise ratio (PSNR) are employed as performance indicators while implementing convolutional autoencoders. Using standard image codecs (such as JPEG and JPEG2000), the current bits for each non-zero quantized transform coefficient are spread across the file. When compression performance increases, the use of high quantization steps will result in a drop in bits per pixel (BPP), which will lead to noise in the reconstructed image. In order to overcome these issues, Ghanbari et al.[10] present certain research that used the denoising approach to improve the quality of the reconstructed image. Zhai et al.[11] suggested an efficient deblocking technique for JPEG pictures that involves post-filtering in shifted windows of image blocks. Foi et al.[12] developed an image deblocking filtering method using shape-adaptive DCT. However, there was also a noticeable acceleration in the development of picture compression techniques. According to Zhang et al.[13], deep learning models have been applied lately to efficiently represent images in order to achieve better results. Convolution neural networks have been employed for image super-resolution (SR) to effectively train deeper networks, particularly when residual learning and gradient-based optimization techniques[14–16] are used. It is possible to improve performance by using an entropy loss rather than an MLP loss, but doing so is more difficult because entropy calculations need the entropy of the error picture. In order to estimate the entropy that can be utilized for density estimation, Erdogmus et al.[17] provided instructions. For GAN networks, we employed two components as the loss function: an adversarial loss to enhance the realism and sharpness of the images, and a first element to measure the similarity between the original and rebuilt images. As an alternative to the Euclidean distance, we might investigate other metrics for measuring similarity. To minimize the Euclidean error in a feature space rather than the original space, a third component can also be included. Alexnet[18] or VGG[19] are two examples of well-known networks that can build the feature space. To obtain better photos for super resolution applications, this technique is employed.[20] Moreover, we can reduce the quantity of redundant data we stored by using the entropy of the compressed representations in the loss function. Since the derivative of the entropy function is zero everywhere but at integers, we can use a smooth approximation of it. We can backpropagate the loss using this approximation.[21] Finally, to encode and decode the images, we employed deterministic autoencoder in each of our networks. Due to their generating capacity, variational autoencoders have been the subject of some investigations, even though we think deterministic autoencoders are more suitable for our situation.[6,22] We can investigate SVD method to compess medical images based on region of intrest.[23] This work proposed to encrypt medical images to enhance the security.[24] The denied conflict with spurious node (TVDCSN) approach was created for wireless communication technologies by block chain-driven Transaction Verification mechanism work to identify malicious nodes and prevent assaults.[24] Kiran P et al.[25,26] proposed region of interest medical image encryption utilizing several maps and a block cypher. The main tool used to extract Region of Interest (ROI) regions is a Laplacian edge detection operator. Special encryption techniques and approaches are usually needed to hide the information in medical photographs.

Methodology

The following section explained about proposed different image compression using machine learning models. Which includes lossless and lossy compression techniques.

Lossless Image Compression

Lossless image compression is crucial for fields like astronomy and medicine that depend on accurate imaging. In this work, we looked into using MLP networks for predictive coding.

Predictive Coding with MLP

Predictive coding (PC) is at the core of most current lossless image compression methods, such as JPEG-LS. Assuming that some of the neighboring pixels are known, the PC attempts to estimate the new pixel value in the image. We are reading the image in raster scan format (first row, second row, etc.) for this project. Thus, we can presume that the green-colored pixels in Figure 2 are known, and the value of "x" is being estimated. To ensure lossless compression, we must store the estimation error in that pixel after the estimation stage. Consequently, we only need to keep a few beginning pixel values to begin the procedure and create the error image, and PC codes the entire image pixel by pixel. The error image will be the same size as the original, but according to information theory, it will contain less entropy. Thus, by employing variable length coding (VLC) techniques, we can reduce the error image's size. We employed Huffman coding for VLC.

PCs have utilized MLP-based networks; however, since the publication of that study, several extensions (such as deeper networks and rectified linear units, or ReLUs) have been shown to be successful in MLP networks. Thus, we attempted to use cutting-edge methods to the implementation of an MLP network for PC. We present our completed network in Figure 3. ReLUs were employed to add nonlinearity after each layer. Be aware that this network is a regression one. We are quantizing the prediction to the closest integer value at the end. We are constructing image-specific weights using this approach, but we can save the network weights without significantly increasing the file size because our layers are short.

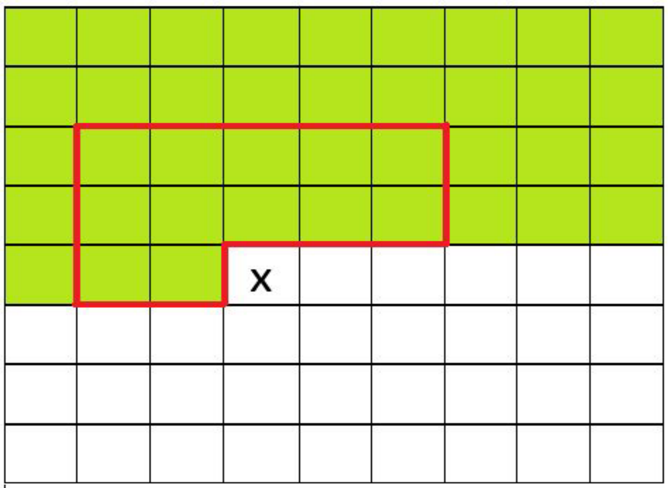


Figure 2. Pixel analysis of Predictive coding for lossless image compression.

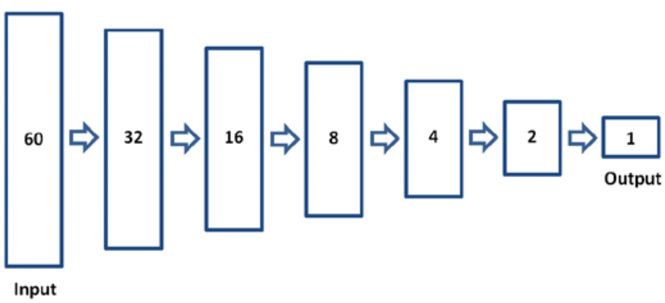


Figure 3. Number of layers used in the multilayer perceptron for image compression

Lossy image compression

Since the human eye cannot distinguish even minute variations in pixel values, a number of picture compression methods have been developed in an attempt to further reduce image size while preserving as much information as possible. The majority of cutting edge compression algorithms use block based techniques since the human eye is less sensitive to local information in images. A basic example of picture patching is shown in Figure 4. Additionally, we experimented with block-based compression in this work, employing 32x32 blocks and several autoencoders with a GAN architecture for lossy compression.

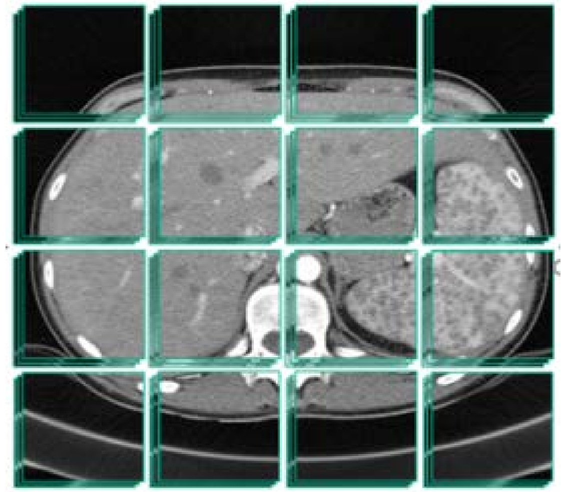


Figure 4. Illustration for block based methods

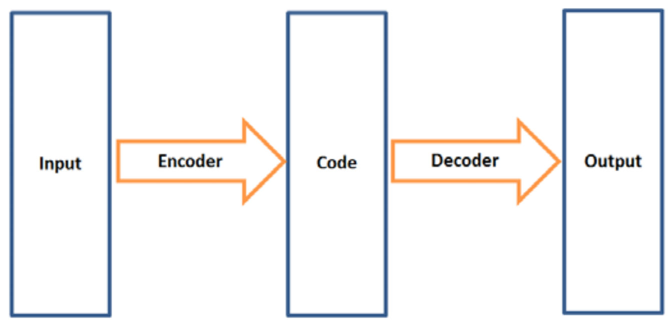


Figure 5. Autoencoder illustration

Compression using autoencoders

Networks known as autoencoders (AEs) attempt to represent an image with less information than the original signal. Dimensionality reduction can be used to accomplish this operation, but we need to ensure that, with only a tiny inaccuracy, we can recover the original signals from the reduced dimension.

Consequently, AE consists of two networks: an encoding network and a decoding network. Latent representation of the original signal is another term for the encoded signal. Figure 5 shows the general layout of the autoencoders. We first looked into MLP-based autoencoders for lossy compression, utilizing vectorized pictures as the autoencoders' inputs. However, we were unable to obtain satisfactory results. This was mostly caused by the absence of spatial information in the image vector representation. Next, we experimented with CNN-AE, or fully convolutional autoencoders. Here, the primary goal is to use max pooling procedures to reduce the image's size. The decoder network is where things become tough. For that section, we have to utilize the deconvolution operator; however, since the convolution operator is not injective, there is no inverse operator. On the other hand, Long and colleagues presented a deconvolutional neural network approximation.

We adopted their strategy, using an interpolation function for deconvolution and a convolution operator afterward. The general design of our convolutional autoencoder, optimized based on our studies, is depicted in Figure 6. We attempted two more improvements to the CNN-AE baseline approach for autoencoder architectures. The first one is predicated on the really basic notion of appending a recurrence relation to the network, as stated in [7]. Since lossy compression is our goal, some inaccuracy will be there in the final product. We attempted to reduce this error by using a recurrent convolutional autoencoder (CNN-RNN-AE), which compresses the error by treating it after each step as a new picture and applying CNN-AE once again. Figure 7 depicts this concept. With this method, the compression ratio and image quality are subject to trade-offs. We employed block-based compression in the test images and trained all of the networks on a database of 32x32 images. We came to understand that our issue is distinct from traditional machine learning challenges, which seek to determine the unknown outcome given a set of inputs. In this case, the input and output are the identical, well-known image from the start. Therefore, while compressing the image, we might be able to use some unique information. Using the patches from the current image to fine-tune the entire network is one straightforward concept. That will, however, provide network weights unique to each image, which we must keep alongside the image. This method does not appear to be practical because deep neural networks have an excessive number of weights. Rather, we choose to fine-tune just the decoder for the provided image, using a very limited decoder architecture. In this manner, we can retain the image-specific data without significantly enlarging the file size. Figure 8 depicts this concept.

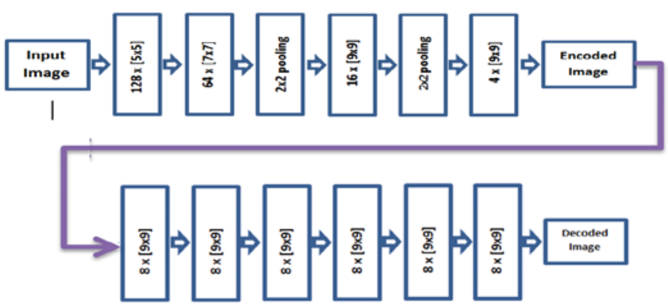


Figure 6. Convolutional autoencoder for encoding and decoding of image.

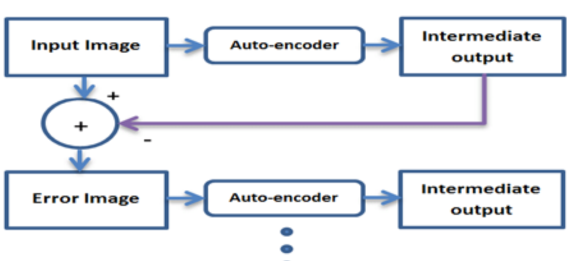


Figure 7. Recurrent CNN-AE for introducing a trade-off between image quality and compression ratio.

COMPRESSION USING GAN

We then posed another query after taking into account the compressed image as an alternative representation of the original image in the latent space. In order to extract more useful information, how can we build the latent representations more effectively? This issue in our situation relates to creating an improved loss function for mapping and inverse mapping the photos. Given that GANs[8] are renowned for producing realistic-looking images, they are an attractive choice for our purpose. To avoid creating a different, rather than realistic, image, we must preserve the necessary information at the same time. The GAN generator that is utilized as a decoder is shown in Figure 9.

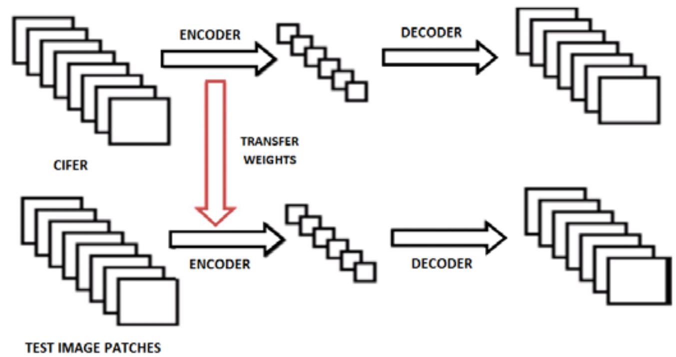


Figure 8. Fine-Tuning approach used for images specific compression

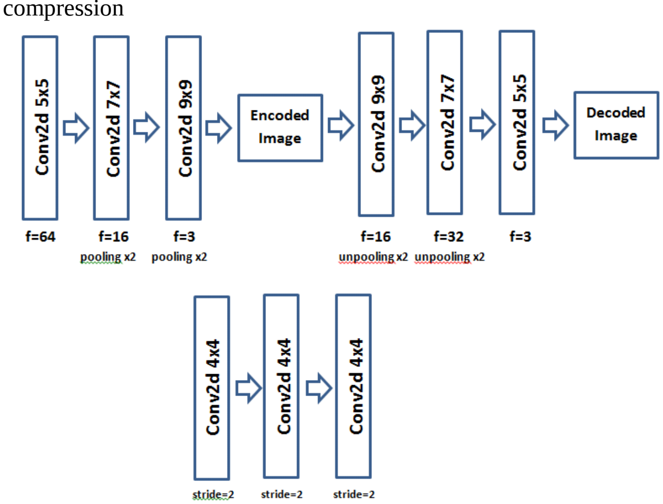


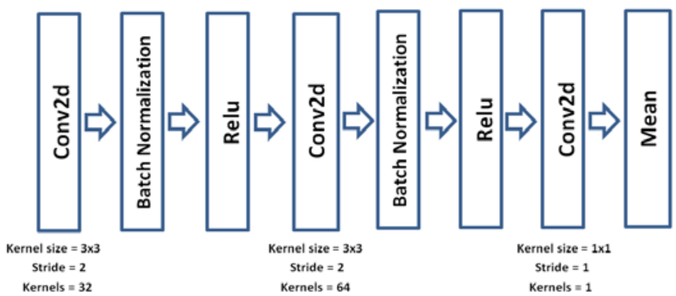
Figure 9. Generator of GANs being using as a decoder

The GANs' generator component can function as a compression decoder. As a result, we chose our latent representations as the image's downsampled version first. To perform reconstruction, we merged the adversarial loss with the mean squared error. The loss function selected for the initial GAN compression network is provided by Equation (1) and is as follows:

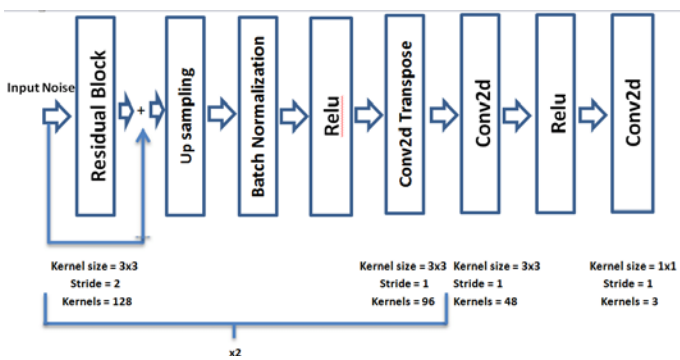
$$L_{total}(x,x') = \lambda_1 L_{error}(x,x') + \lambda_2 L_{adversarial}(x,x',w,\theta) \qquad (1)$$

where the weights of the generator and discriminator are denoted by θ and w, respectively. Our adversarial loss function of choice was the Wasserstein loss (WGAN loss) function. Since the classical GAN loss minimizes the Jensen-Shannon distance, which is undefined when the probability densities are in a low-dimensional manifold as they are in our situation, it is unstable. The error function produced by the Wasserstein loss is more stable and convergent, making interpretation simpler and maybe yielding better outcomes. Regarding the application, we cut the weights, deleted the sigmoid layer, and removed the logarithm from the loss function. The reader is advised to read for further information.

The second method is to obtain the compressed form by substituting an encoder output for the downsampled images. By including an encoder-decoder scheme, the system can choose its own latent representations. As seen in Figure 10, the autoencoder and the GAN structure can be trained simultaneously or sequentially. We put the simultaneous training into practice. We assessed the network using the L1 or L2 norm as the second term and the classical GAN or WGAN loss as the first. To determine a suitable weighted balance between these two terms, we validated the network. Figure 11 depicts the CNN-AE-FT decoder and encoder network.



(a)



(b)

Figure 10. Compression achieved using combination of both auto encoder (a) and GAN (b)

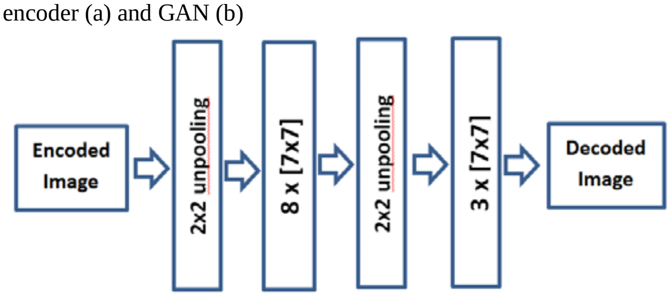


Figure 11. Decoder network for CNN-AE-FT. Encoder part is same with CNN-AE

Dataset and Metric

We used three test photographs for training and testing in our research, following a methodical methodology that was customized for our image-specific lossless compression method. Bits per pixel (bpp) is a trustworthy statistic for evaluating lossless compression performance, and we used it to evaluate the efficacy of our method. We used the CIFAR dataset, which is well-known for having a large collection of 50,000 32x32 RGB images, to train our model for lossy compression. This dataset is noteworthy since it includes 50,000 samples in the training set and 10,000 samples in the test set, providing extensive coverage for training. In addition, we went beyond CIFAR to include a dataset similar to CIFAR-10, but with 100 different classes, each with 600 photos. With the addition of new difficulties such as noise, orientation, and image dimensions, this enlarged dataset provided a more thorough evaluation framework. We carried out preprocessing actions targeted at normalizing image sizes in order to guarantee fair comparisons across various compression techniques. In order to achieve consistent processing, the original 32×32 images were enhanced by adding four zero pixels on each side. Then, to replicate real-world situations where photos may experience varying degrees of compression during transmission or storage, the enhanced images were subjected to random compression with a chance of 0.5. These painstaking methods were necessary to confirm the stability and effectiveness of the compression strategies we suggested.

The following three factors led us to select this dataset: It is reasonable to employ patch-based compression on high resolution (HR) images to preserve local relationships in the pixel values because (i) it contains a large number of samples; (ii) it consists of small images, therefore training a network on CIFAR does not take days.

We also tested the actual performance of our approaches using some HR photos. Our algorithms were trained on the CIFAR dataset for the HR images, and the resulting weights were then applied to the HR image patches. Two distinct metrics were employed in our comparisons. Equation (2) can be used to formulate the peak signal to noise ratio (PSNR), which is the first one. Decibels (dB) are used to measure peak signal-to-noise ratio (PSNR), which is used to evaluate reconstruction error. Equation (3) provides the mean square error (MSE), which defines it as follows.

$$PSNR = 20\log_{10}\left[\frac{255}{\sqrt{MSE}}\right] \quad (2)$$

Reconstruction of an image or signal is best when the reconstruction errors are small (around zero). Therefore, during signal/image reconstruction, MSE minimization (PSNR maximization) is crucial.

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - I'(i,j)]^2 \quad (3)$$

After applying the encoder and decoder, I is the reconstructed image. I is also the input image. Keep in mind that higher PSNR values indicate greater algorithm performance. Nevertheless, they demonstrated that because PSNR does not account for HVS, it is not a useful comparison statistic for images [12]. The structural similarity index (SSIM), a novel metric, was put out by them. Equation (4) can be used to formulate it.

$$SSIM(x,y) = \left[\frac{(2\mu x\mu y+C1)}{(\mu x^2+\mu y^2+C1)} \frac{(2\sigma xy+C2)}{(\sigma x^2+\sigma y^2+C2)}\right] \quad (4)$$

where L is the dynamic range of pixel values (255 for 8 bit unsigned integers), k1 = 0.01 and k2 = 0.03 by default, μ X is the mean value of X, σ X is the standard deviation of X, and σ XY is the covariance of X and Y. After calculating SSIM on small image windows, the average is applied to the entire image. The decimal value that emerges is between -1 and 1. Since SSIM=1 indicates that all of the images are identical, we want it to be as near to 1.

EXPERIMENTS

MULTI LAYER PERCEPTRON (MLP)

Using the approach depicted in Figure 12, we coded the error images for the lossless compression using Huffman coding. The size of the prediction block is one of the key variables in predictive coding. It becomes more difficult to predict the new pixel value and employ a deep network for tiny box sizes when we expand it too much, as the pixels become uncorrelated with the subsequent pixel. Following various experiments, we selected 60 pixels as the prediction size. We employed a straightforward MSE loss as the loss function.

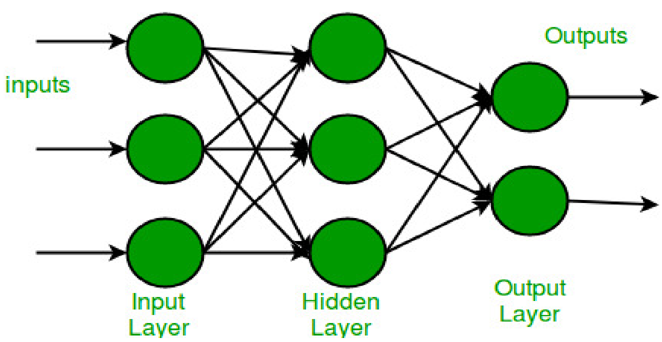


Figure 12. Architecture of multilayer perceptron network

An output layer, one or more hidden layers, and at least one input layer are the three distinct functions that make up a feed forward neural network, or MLP. Both the hidden and output layers contain a large number of neurons with activation functions. Neural Networks can be meticulously designed to simulate intricate non-linear interactions between input parameters and the result. Selecting the activation function and the quantity of hidden layers and neurons in each layer are a few examples of this. Creating a good NN architecture for a given task is not simple. Tests are typically employed to determine the optimal configuration for a neural network because there is no standard technique for calculating the optimal number of neurons and layers.

AUTOENCODERS

All of our fully convolutional models are effective autoencoders. In our research, we consistently employed a 16:1 compression ratio for consistency and ease of comparison with JPEG. The most crucial aspect of autoencoders is that, because we will be storing this representation as a compressed image, we cannot employ an excessive amount of filters in the final layer prior to the latent space. Due to this difficulty, we had to employ asymmetric autoencoders instead of symmetric ones. The performance of really deep networks is not very good because our input images are 32x32. Following a series of trials, we determined that the network shown in Figure 6 was optimal for CNN-AE and CNN-RNN-AE. We employed three recursive phases in CNN-RNN-AE. Although CNN-AE-FT includes about 1300 weights of 32-bit floating integers in the decoder, we can use the same architecture for it as well. However, storing them with the compressed picture will be inefficient. Therefore, we made the decision to fine-tune our technique using a shallower decoder. We employed the Adam optimizer in all AE variants, using MSE as the loss function and a learning rate of 10-5.

GENERATIVE ADVERSARIAL NETWORKS (GANS)

A discriminator network plus a decoder network make up the first GAN network. We took inspiration for the decoder network from Resnet [13], however instead of making it as deep as possible because our images are 32 by 32, we wanted to manipulate the parameters more readily and have a shorter training time because of time restrictions. The residual, up sampling, batch normalization layer, relu, conv2d, and conv2d transpose blocks make up the decoder (generator). We upsample twice to return to the original size of the convolutional layers in the decoder section, which include paddings and a stride size of 1. All that makes up a discriminator are convolutional layers. To reduce the saturation of the activation units in convolutional layers, we employed Xavier initialization.[14].

As recommended in the paper [9], we utilized RMSProp, clipped weights between [A-0.01, 0.01] interval, and took 2.10 −5 as the learning rate for the Wasserstein loss function. With λ1 = 1 and λ2 = 10, the SSIM findings are optimal. Since that term minimizes MSE directly, the MSE error reduced as expected when we increased λ2. We utilized an ADAM optimizer with beta = 0.5 and a learning rate of 2.10−4 for the DCGAN loss.

A discriminator plus an auto-encoder make up the second GAN network. For each loss function, the same learning rate and coefficients were employed. Convolutional networks with max pooling and upscaling layers are used as encoders and decoders to regulate the image's size.

RESULTS AND COMPARISON

Although there are many different compression methods that claim to be superior to JPEG, the most of them don't include implementation details, and JPEG is still the most used algorithm for both lossy and lossless compression. Therefore, we shall compare our findings with JPEG and JPEG-2000 in this study.

Medical image compression techniques are investigated in this work. Hence, the most prevalent types of medical images—such as CT, MRI, ultrasound, and PET scans—are selected to serve as simulation examples.[27] All of these simulated samples' key features are listed in Table 1.

Table 2 compares the bit-per-pixel (bpp) of JPEG, JPEG-2000, and our network. Thus, our network performs significantly better than JPEG and approaches JPEG-2000 in terms of performance. Also take note of the fact that modern smartphones and PCs can operate this MLP network in real time. Figure 13 shows the sample medical images used for experimental anlysis.

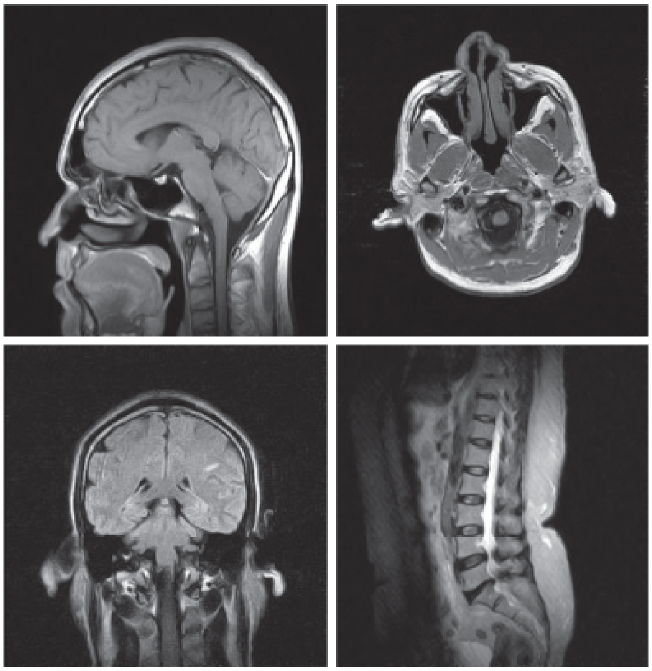


Figure 13. Sample Medical images used in the proposed work

Table 1. Provide examples of simulation image sample.

Image ID	Image Type	Body part	Image Size
1	Ultrasound	Fetus	256*256
2	MRI	Brain	256*256
3	X-ray	Hand	256*256
4	X-ray	Leg	256*256
5	X-ray	Knee	512*512
6	X-ray	foot	512*512

Table 2. Comparison of average bpp rates for different lossless compression algorithms on the test images.

Algorithm	Bits per Pixel
JPEG	6.3
JPEG-2000	3.75
MLP	4.53

We compared our findings for the lossy compression utilizing patch-based compression on both CIFAR test data and HR test photos. We displayed the findings for our top-performing CNN-AE and GAN networks in Tables 3 and 4. Every outcome is documented with a compression ratio of 16:1. We would want to notify the reader that because we did not incorporate entropy coding into our algorithms, these comparisons are not entirely fair. All of our methods beat JPEG on the CIFAR test data, with the exception of GAN-AE with L1+DC loss. Our best-performing network is GAN-AE with L2+Wloss. It is evident that WGAN outperforms DCGAN, and that performance is further enhanced by the addition of L2 loss. With L1 loss, we were unable to perform well. The outputs got sharper and more attractive as we increased the weight of the adversarial loss, but we ran into some color-related issues. The network occasionally deteriorated and needed more rounds to achieve the same color with the original image. However, these issues are reduced after appropriate network parameters are identified, and the reconstructed image is crisper and of higher quality. Additionally, CNN-AE and CNN-RNN-AE both had excellent performances.

Table 3. Result of various architecture on CIFAR dataset.

CIFAR	
Methods	PSNR
JPEG-2000	42.5
GAN-AE (L2,W)	35.78
GAN-AE (L2, DC)	20.54
CNN-RNN-AE	30.4
GAN WGAN	29.96
CNN-AE	29.5
GAN-DC	29.45
GAN-AE (L1,W)	29.04
JPEG	29.24
GAN-AE (L1,DC)	29.23

Table 4. Result of various architecture on HR dataset.

HR Images		
Methods	PSNR	SSIM
JPEG-2000	35.1	0.992
GAN-AE-FT	33.7	0.99
GAN-AE (Best)	32.51	0.985
JPEG	32.4	0.976
CNN-RNN-AE	29.3	0.971
CNN-AE	28.6	0.963

None of our methods perform at a level that is equivalent to JPEG-2000 in the CIFAR data. On the HR image set, patch-based comparison methods allow us to obtain results that are comparable when using JPEG-2000. Table 3 has the results for HR images. Once more, GAN-AE performed incredibly well. Here, we have simply presented the top-performing GAN-AE network using the Wasserstein metric and L2 loss. SSIM indicates that CNN-AE-FT outperforms GAN-AE network and appears to be highly successful, with an index that is quite near to JPEG-2000. It demonstrates the huge promise of image-specific compression through fine-tuning. Observe also the improvement in JPEG performance between CIFAR and HR pictures.

In order to evaluate the visual image quality, the effectiveness of the suggested model is lastly tested on several frames in terms of PSNR. The comparative findings are shown in Figure 14.

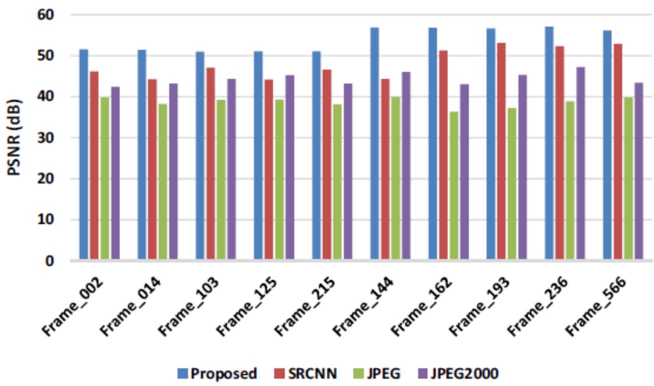


Figure 14: Comparative analysis of proposed work in terms of PSNR.

Conclusion and Future work

We have shown many networks in this study to compress medical images with and without good resolution. Our findings are based on the network configurations and settings that we experimented with. While the majority of our results show promise, they also suggest numerous routes and possible changes. We attempted a very basic architecture for the lossless compression and concentrated on the lossy compression problem in our article. We used MSE loss to train the MLP network. It is possible to improve performance by using an entropy loss rather than an MLP loss, but doing so is more difficult because entropy calculations need the entropy of the error picture. We can utilize density estimation techniques like Parzen windowing to estimate the entropy. For the GAN, we employed two elements as the loss function. As an alternative to the Euclidean distance, we might investigate other metrics for measuring similarity. To minimize the Euclidean error in a feature space rather than the original space, a third component can also be included. Since the proposed DWT–CNN paradigm may be employed with current image coding standards like JPEG, JPEG2000, or BPG, it is highly suitable for a variety of procedures. The testing results across many performance indicators confirmed the superiority of the proposed model over state-of-the-art methods. Additional sources, like the representation in a feature space, can be employed to retain more data. Additionally, we can compress depending on the class information and condition on the classes, which might produce better outcomes. One option is to employ a fine-tuning strategy. The storing of the decoder weights is minimal in videos since images can share the decoder weights for the successive collection of frames. GAN fine-tuning is another option. Beyond real-time hardware execution, the suggested method can be used to any other applications in the future. Various deep learning approaches could be applied in the future to spare data processing complexity and reduce band redundancy for 5G network transmission and storage. Future performance of the proposed model can be enhanced by modifying the CNN model's hyperparameters, which include the number of hidden layers, learning rate, epoch size, and other variables.

Conflict of interest statement

Authors declare that there is no conflict of interest for publication of this work.

References and notes

1. C.E. Leiserson, N.C. Thompson, J.S. Emer, et al. There's plenty of room at the top: What will drive computer performance after Moore's law? *Science (80-.).* **2020**, 368 (6495), 9744.
2. M. Nanavati, M. Shah. Performance comparison of different Wavelet based image fusion techniques for Lumbar Spine Images. *J. Integr. Sci. Technol.* **2024**, 12 (1 SE-Biomedical and Pharmaceutical Sciences), 703.
3. C. Perera, C.H. Liu, S. Jayawardena. The Emerging Internet of Things Marketplace from an Industrial Perspective: A Survey. *IEEE Trans. Emerg. Top. Comput.* **2015**, 3 (4), 585–598.
4. B. Kalva, M.C. Mohan. Effective image retrieval based on an optimized algorithm utilizing a novel WOA-based convolutional neural network classifier. *J. Integr. Sci. Technol.* **2023**, 11 (3), 523.
5. G. Toderici, D. Vincent, N. Johnston, et al. Full resolution image compression with recurrent neural networks. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*; **2017**; Vol. 2017-Janua, pp 5435–5443.
6. S. Santurkar, D. Budden, N. Shavit. Generative Compression. In *2018 Picture Coding Symposium, PCS 2018 - Proceedings*; IEEE, **2018**; pp 258–262.
7. Y. Yang, S. Mandt, L. Theis. An Introduction to Neural Data Compression. *Found. Trends Comput. Graph. Vis.* **2023**, 15 (2), 113–200.
8. T.G. Shirsat, V.K. Bairagi. Lossless medical image compression by IWT and predictive coding. *2013 Int. Conf. Energy Effic. Technol. Sustain. ICEETS 2013* **2013**, 2013, 1279–1283.
9. N. Johnston, D. Vincent, D. Minnen, et al. Improved Lossy Image Compression with Priming and Spatially Adaptive Bit Rates for Recurrent Networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*; **2018**; pp 4385–4393.
10. M. Ghanbari. Standard codecs: Image compression to advanced video coding (3rd edition). *Standard Codecs: Image Compression to Advanced Video Coding*. Iet 2011, pp 1–482.
11. G. Zhai, W. Zhang, X. Yang, W. Lin, Y. Xu. Efficient image deblocking based on postfiltering in shifted windows. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, 18 (1), 122–126.
12. A. Foi, V. Katkovnik, K. Egiazarian. Pointwise shape-adaptive DCT for high-quality deblocking of compressed color images. *Eur. Signal Process. Conf.* **2006**, 16 (5), 1395–1411.
13. K. He, X. Zhang, S. Ren, J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*; IEEE Computer Society, Las Vegas, NV, USA, **2016**; Vol. 2016-Decem, pp 770–778.
14. J. Duchi, E. Hazan, Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *COLT 2010 - 23rd Conf. Learn. Theory* **2010**, 12 (7), 257–269.
15. T. Takase, S. Oyama, M. Kurihara. Effective neural network training with adaptive learning rate based on training loss. *Neural Networks* **2018**, 101, 68–78.
16. D.P. Kingma, J.L. Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. 2015.
17. D. Erdogmus, J.C. Principe. Entropy minimization algorithm for multilayer

perceptrons. *Proc. Int. Jt. Conf. Neural Networks* **2001**, 4 (01CH37222), vol. 4), 3003–3008.

18. A. Krizhevsky, I. Sutskever, G.E. Hinton. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, 60 (6), 84–90.
19. K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. 2015.
20. C. Ledig, L. Theis, F. Huszár, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*; **2017**; Vol. 2017-Janua, pp 105–114.
21. S. Fraihat, M.A. Al-Betar. A novel lossy image compression algorithm using multi-models stacked AutoEncoders. *Array* **2023**, 19, 100314.
22. K. Gregor, F. Besse, D.J. Rezende, I. Danihelka, D. Wierstra. Towards conceptual compression. In *Advances in Neural Information Processing Systems*; **2016**; pp 3556–3564.
23. Kiran, B.D. Parameshachari, D.S. Sunil Kumar, P.S. Prafulla, J. Yashwanth. Singular Value Decomposition (SVD) Based Optimal Image Compression Technique. In *2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques, EASCT 2023*; IEEE, **2023**; pp 1–6.
24. K. Prabhavathi, M.B. Anandaraju, Kiran. An efficient medical image encryption algorithm for telemedicine applications. *Microprocess. Microsyst.* **2023**, 101, 104907.
25. Kiran, B.D. Parameshachari, H.T. Panduranga. Medical Image Encryption Using SCAN Technique and Chaotic Tent Map System. In *Recent Advances in Artificial Intelligence and Data Engineering*; Springer, Berlin/Heidelberg, Germany, **2022**; pp 181–193.
26. P. Kiran, B.D. Parameshachari. Resource Optimized Selective Image Encryption of Medical Images Using Multiple Chaotic Systems. *Microprocess. Microsyst.* **2022**, 91, 104546.
27. A.S. Chauhan, J. Singh, S. Kumar, et al. Design and assessment of improved Convolutional Neural Network based brain tumor segmentation and classification system. *J. Integr. Sci. Technol.* **2024**, 12 (4 SE-Computer Sciences and Mathematics), 793.