Fayoum University Faculty of Engineering, 2025, Vol: 8(2), 1-11
https://fuje.journals.ekb.eg/
ISSN Online: 2537-0634
ISSN Print: 2537-0626

Efficient Image Compression Technique for Low Bit Rate Requirements

Hatem Taha[1,*], Lamiaa A. Elrefaei[2], Haitham Akah[3], Maher Abdelrasoul[2]

[1] Payload Department, Egyptian Space Agency, Cairo, Egypt
[2] Electrical Engineering Department, Faculty of Engineering at Shoubra, Benha University, Cairo, Egypt
[3] Space Communication Department, Egyptian Space Agency, Cairo, Egypt

*Corresponding author: Hatem Taha (Hatem.taha@egsa.gov.eg).

How to cite this paper: Taha, H., Elrefaei, L.A., Akah, H. and Abdelrasoul, M. (2025) Efficient Image Compression Technique for Low Bit Rate Requirements, *Fayoum University Journal of Engineering, Vol: 8(2), 1-11*.
https://dx.doi.org/10.21608/fuje.2025.314170.1091

Abstract

Recently, there has been a surge in learned image compression techniques that outperform standard image compression algorithms, leading to a growing interest in the application of neural networks for effective image compression. Most of these new techniques prioritize improving image quality, without adequately considering processing power, processing time and system resource constraints. In this paper, we propose a promised autoencoder-based lossy image compression method. Our aim is to develop a compression technique that achieves good compression performance while minimizing computational complexity. The method we propose involves partitioning the input image into two distinct groups according to the content of image blocks. For each group, we employ separate autoencoders to achieve a reduced bit rate while enhancing image quality. Subsequently, a quantization process is applied before entropy encoding to get the final outputs. The experimental results demonstrate that our proposed method achieve better PSNR than JPEG compression in low bit rates.

Keywords

Image compression; Neural networks; Autoencoder; Bit rate; PSNR

1. Introduction

Image Compression is used in various domains and applications, it plays an essential role in managing and transmitting visual data efficiently [1], it is almost used everywhere like Digital Cameras, advertising, web design, mobile application, social media, gaming, medical imaging, remote sensing and more. Image compression is the process of reducing the size of a digital image file. It aims to minimize the storage space required for the image while retaining its essential visual information. This reduction in file size is achieved by removing redundant or unnecessary data from the image, image compression schemes can be divided into two broad classes [2]:

- Lossless image compression, where the reconstructed image is identical to the original and it involves no loss

of information.

- Lossy image compression, which provide much higher compression than lossless compression but allow the reconstructed image to be different from the original based on the quality.

In recent times, learning techniques have been successfully applied to a large number of computer vision and image processing tasks, some of these learned image compression techniques outperform standard image compression algorithms such as JPEG [3] and JPEG2000[4], that leading to a growing interest in the application of neural networks for effective image compression. The learning objective is thus to minimize the difference between the reconstructed and the original images. The general pipeline of learned technique includes an auto-encoder architecture as represented in [5], [6], where an encoder first compresses the input image into a compact representation, then fed into a decoder to reconstruct the input image. The general architecture of learned image compression is shown in Figure 1.

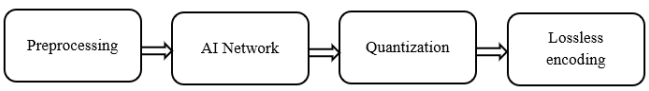

Figure 1. General Pipeline of learned image compression technique.

Typically, the researchers build their structure of neural network for image compression. This network structure undergoes training using specific image datasets until achieving the desired balance between compression ratio and quality. The common part between the encoder and the decoder is the trained network, the encoder compresses the original image utilizing this network. The compressed output is then directed to the quantizer and entropy encoder to obtain the final compressed output. The decoder aims to faithfully reconstruct the original image from the compressed representation received from the encoder, utilizing operations that are the reverse of those applied during the encoding phase. The deep Convolutional Neural Networks (CNN) based algorithms like VGG-16 **Error! Reference source not found.** combine from thirteen convolutional layers, five Max Pooling layers, and three Dense layers which sum up to 21 layers and AlexNet **Error! Reference source not found.** have 5 convolutional layers and 96 filters in the first layer and ResNet-50 [34] have 50 convolutional layers and 96 filters in the first layer. The CNN approach holds an advantage in transmitting highly compact data while achieving high-quality reconstructed images, relying on correlations between the original image and the dataset used for training. However, these methods often demand extensive processing power and time, making them unsuitable for numerous applications due to these resource-intensive requirements.

In applications like drone imaging and remote sensing satellites, the utilization of rapid previews or image placeholders is indispensable. The image compression plays a crucial role to introduce a quick preview or general prospective for images where timely access to visual information can make a significant difference in outcomes and efficiency. Here are some key reasons that highlight the quick preview value:

Rapid Decision-Making: In applications where time is critical, such as emergency response, military operations, or disaster monitoring, quick previews allow stakeholders to make initial assessments and decisions without waiting for full-resolution imagery. This can be crucial for saving lives and resources.

Data Assessment: Quick previews provide a way to assess the quality and relevance of collected data before committing to extensive processing or analysis. Users can quickly determine if the data meets their needs.

Resource Efficiency: Storing and transmitting high-resolution images or data can be resource-intensive in terms of bandwidth and storage. Quick previews are typically smaller in size, making them more efficient for initial data handling.

Remote Sensing: In remote sensing applications, quick previews can assist in identifying areas of interest or changes in the environment, helping researchers or analysts focus their attention on specific regions.

Low-Bandwidth Environments: In regions with limited internet connectivity or remote areas where bandwidth is

scarce, quick previews are essential for accessing at least some information even when high-resolution data may be challenging to obtain.

In this paper, we introduce an innovative image compression method employing a low complexity autoencoder, specifically designed to reduce computational requirements. This approach holds great potential for systems with limited resources that require a low bit rate and low computational complexity for applications that need general prospective or quick previews, while maintaining an acceptable level of image quality. However, it is important to evaluate the trade-offs between compression performance and computational complexity to determine the suitability of this method for specific applications. The rest of this paper is organized as follows: In Section 2, we present the related work. In section 3, we present the idea and the typical architecture for our proposed method and describe the effect of autoencoder main parameters (activation function, number of layer and latent size) on the mean square error of reconstructed image then present the chosen autoencoders for applying our method. In Section 4, we present the evaluation metrics, and the results of our proposed method compared to JPEG compression. Finally, in Section 5, we conclude the advantage and disadvantage of the proposed method.

2. Background and Related Work

Neural network-based image compression autoencoders are compression models trained to compress images from defined image dataset, aiming to transform the input images into outputs while keeping distortion levels as low as possible [10]. Many state-of-the-art neural codecs and compression algorithms are derived from or inspired by the principles of compressive autoencoders [11], [12] and [13]. Several deep learning autoencoder architectures are proposed for image compression tasks, each offering unique approaches and features. These differences primarily emerge in how they handle preprocessing, choose loss functions, configure network layers, employ activation functions, implement quantization techniques, and incorporate lossless encoding mechanisms. For example, the authors in [14] design a symmetric deep convolutional auto-encoder based image compression (CAE) architecture with multiple down-sampling and up-sampling units to replace the conventional transforms. In [15], the authors propose a three-layer image compression consisting of a base-layer versatile video coding (VVC) (intra) codec, a learning-based residual layer codec, and a learnable hyper-prior. The contribution in that study was developing a data fusion attention module and integrating several known components together to form an efficient image codec, which has a higher compression performance than the standard VVC coding scheme. In [16], the authors propose an enhanced Invertible Encoding Network with invertible neural networks (INNs) to largely mitigate the information loss problem for better compression. Their method outperforms the existing learned image compression methods and compression standards. In [17], the authors introduce a method that is combined from two decoder, the main decoder is a model for decoding entire images and the second decoder (selective detail decoder) is a model specialized for specific important parts and trained using weighted MS-SSIM decoders with keeping a single encoder. In [18], the authors introduce a new method for image quality preservation based on the concept of adaptive thresholding and quantization using image content characteristics. Experimental results show that overall image quality preservation is noticeably improved over the Q-factor approach and the MPEG2 adaptive quantization algorithm at the same compression levels.

Also, many researchers employed the Variational Autoencoder (VAE) for image compression. In [19], the authors use VAE with the Challenge on Learned Image Compression (CLIC) dataset for training the model. The model was enormous and complex due to multiple training parameters. In [20], a VAE-based architecture for high resolution image compression is proposed. It utilizes a non-local attention module to improve the training process, but at the cost of increased model complexity. In [21], the authors proposed a VAE for image compression, and in [22], VAE was used to compress images and achieve a 4.10 bits per

pixel rate, but with a complex model architecture.

Various modern approaches that apply deep learning on image compression tasks can indeed produce better results compared to traditional or standard compression techniques. However, achieving these superior outcomes often demands significant computational resources, including extensive processing power, large memory requirements, and longer processing times [23], [24] and [25].

One alternative approach for developing an efficient learned-based image compression system involves employing lightweight architectures and optimization techniques. In [26] a Stacked Autoencoder (SAE) model was introduced as an image compression model, SAE consists of stacked neural network layers fully connected, where each layer's output is the input for the next layer. Results show that their proposed SAE model outperform JPEG compression algorithm for MNIST images dataset, but it has a performance less than JPEG for the greyscale and color images datasets. In [27], an Implicit Neural representation approach was presented for image compression. The encoding process involves an overfitting training a Multilayer Perceptron (MLP) for the image, followed by quantizing its weights for transmission. Upon decoding, the received MLP is applied to all pixel positions to reconstruct the original image. This approach offers a notable advantage in terms of model size, which tends to be significantly smaller compared to conventional learned-based compression algorithms. However, it comes with limitations. Encoding tends to be slow, and when evaluated against state-of-the-art compression methods, this approach demonstrates less effective performance. The authors in [28] and [29] introduced modifications to the approach outlined in [27], resulting in enhancements to the method and achieving improved performance compared to the original proposal. These modifications likely aimed to address the limitations or shortcomings observed in the initial method, potentially refining the encoding or decoding processes, optimizing model architecture.

Most of these new techniques prioritize improving image quality or reducing the bit rate, without adequately considering processing power, processing time and system resource constraints. Some application such as small drones and Cube-Satellites cannot keep up with the exhaustive needs of modern image compression approaches. Further, some applications do not need high quality images but need low compression size and high speed.

3. Proposed Method

Content-based compression takes into account the importance of different regions of an image, unlike traditional compression methods that treat the entire image uniformly, content-based compression focuses on identifying and maintaining the perceptually important regions, while compressing less significant regions more aggressively. This approach is particularly useful when certain parts of an image are more crucial than others. The primary goal is to allocate more bits and compression resources to preserve the details and features that are considered visually important, while allowing less important regions to be compressed more aggressively.

3.1. Proposed Architecture

Our method is a type of content-based compression, it is rooted in the recognition that a single natural image contains varying levels of details. This variability in detail levels within an image serves as the foundational concept driving our approach. The proposed method involves dividing the image into (8x8) pixel blocks. Through this segmentation, certain blocks exhibit higher levels of detail compared to others. Our approach is to classify these blocks into two distinct groups: Group 1 (G1), encompasses blocks with lower detail levels, and Group 2 (G2), encompasses blocks characterized by higher detail levels. This separation is based on the varying levels of details found within these blocks, as further elaborated in subsequent sections.

The classification of image blocks is determined by measuring the Mean Squared Error (MSE) between the

inputs and outputs G1 autoencoder, **Error! Reference source not found.** provides a visual representation, outlining the step-by-step process involved in the segmentation and separation of groups within the image blocks.

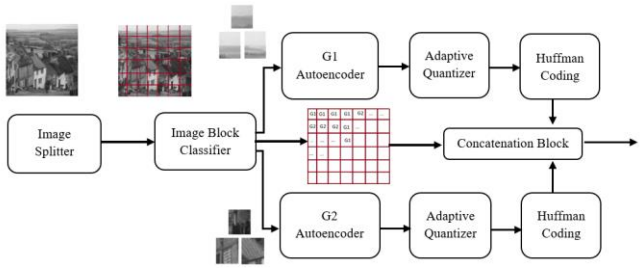

Figure 2. Proposed Method Architecture.

As a result, for image blocks classifying, each block has a sign bit, this bit is used to define the group type whether group1 or group2.

The significance of this bit extends into the decoding process, serves as a crucial identifier, enabling the clear classification of each block, ensuring its placement into either Group 1 or Group 2 during decoding, this ensures the proper reconstruction and arrangement of blocks into their respective groups, facilitating an effective decoding process. Following the completion of block grouping and extraction of outputs, each group's outputs undergo Huffman lossless encoding. This encoding methodology is employed to reduce the final size of the outputs for both Group 1 and Group 2. Huffman encoding, known for its efficiency in compressing data by assigning variable-length codes to symbols. It is utilized here to further decrease the size of the outputs while retaining all the information without any loss.

3.2. Autoencoder Section

An autoencoder is a specific type of neural network, which is mainly designed to encode the input into a compressed and meaningful representation and then decode it back such that the reconstructed input is similar as possible to the original one [30]. The general architecture of an autoencoder includes an encoder, decoder, and bottleneck layer as shown in Figure 3. The encoder and decoder stages contain single or multi-layer and each layer may use activation function or not.

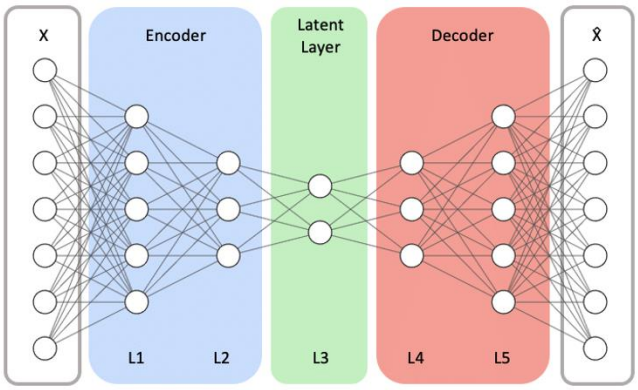

Figure 3. General Architecture of an Autoencoder from [31].

The autoencoders selection experiments consist of two phases: firstly, an experiment setup phase which involves the selection of the most suitable autoencoders; secondly, the training phase which involves the selection of the most fitting weights for both autoencoders.

A series of experiments were carried out to identify the optimal setup for achieving low complexity with efficient compression and quality reconstruction within each group. These experiments explore various configurations and parameters to ascertain the most fitting autoencoder architecture for the compression process.

Several key parameters significantly influence the performance of autoencoders in neural network design. These parameters are pivotal in shaping the effectiveness and efficiency of the autoencoder. The main parameters that are used in our autoencoder designing are:

a) Error and Activation functions
b) Number of layers
c) Output layer

3.2.1. Error and Activation Function

In this experiment, we evaluate different activation functions to select the one which achieve our requirements, the Mean Squared Error (MSE) is used as evaluation metric, it is a straightforward and widely used metric for quantifying the difference between the original image and the reconstructed image as shown in Equation (1). Using MSE as the error function in learned compression algorithms is

a common approach, as highlighted in studies [26], [32] and [33]. In our proposed approach, we have planned to employ MSE as the error function throughout the learning process. The MSE equation is:

$$MSE(x,y) = \frac{1}{n}\sum_{1=1}^{n}(x_i - y_i)^2 \quad (1)$$

- x and y are two images being compared
- x_i and y_i represent the pixel values at the corresponding position in the image
- n is the total number of pixels in each image.

The activation function is used to add type of non-linearity to the AI model, to select the appropriate activation function for our simple autoencoders, we apply different activation function to four-layer autoencoder with four outputs at the middle (latent) layer and 64 pixels input and output layers, table 1 lists the results of our experiment for some of the generic activation functions.

Table 1. Activation function effect on MSE

Activation function	Lena	Boat	Goldhill	Average
elu	90.65	160.13	115.12	121.97
relu	283.52	191.61	191.85	222.33
sigmoid	133.62	172.95	136.1	147.56
swish	92.15	160.29	115.85	122.76
linear	90.86	172.0	119.62	127.5
None (no activation function)	91.3	159.3	114.48	121.7

From Table 1. we notice that the absence of activation function is not affecting the result for this simple autoencoder, it may be useful in other types of autoencoder such as deep learning autoencoders.

3.2.2. Number of Layers

In this experiment, our aim is to identify the hidden layers that are most appropriate for our requirements. Based on activation function experiment results, we will construct our autoencoder without activation function. We built four simple autoencoders with different number of layers (2, 4, 6 and 8) and the same number of outputs nodes (1 node) to define the best number of layers for our autoencoder. The process applied on same images (Lena, Boat and Goldhill). The training process built on anaconda platform using python code and TensorFlow library.

After training process, we notice that; almost there are very small effects for higher network layers on the results (MSE), therefore the 2-layer network is selected for our method. The difference between the four networks as listed in Table 2.

Table 2. Network Layers effect on MSE

Number of layers	Mean Square Error (MSE)				Total Parameters
	Lena	Boat	Goldhill	Average	
2	282.28	386.56	262.67	310.5	128
4	280.93	382.37	263.37	308.89	1040
6	286.16	383.23	261.41	310.27	2320
8	280.56	382.05	286.8	316.47	5392

3.2.3. Output Layer

In this experiment, our aim is to identify the number of outputs that are most appropriate for our requirements. Based on the previous experiments results, we will construct our autoencoder based on two layers (input, output and one hidden layers), while the number of network parameters in this network are very small compared to the other networks, and the processing time is much faster than other networks because the number of arithmetic operations that are used in this network are very limited compared to other networks. We perform the compression on the same images and the tables 3, 4 and 5 list the compression sizes for two approaches at the same reconstructed PSNR.

Table 3. Lena Image Compression Aize for Different Outputs

PSNR (Lena image)	4 Outs	6 Outs	8 Outs
24.2443	18986	19727	20240
26.4703	24990	26655	27722
27.3229	30487	30497	31740
28.2432	-	37082	36941
28.8857	-	-	42102
29.2576	-	-	48195

Table 4. Boat Image Compression Aize for Different Outputs

PSNR (Boat image)	4 Outs	6 Outs	8 Outs
23.3364	22761	24320	24911
25.1009	29843	31280	32571
25.5879	38574	33802	35074
26.0329	-	36718	37483
26.8229	-	47068	43268
27.4565	-	-	52263
27.6304	-	-	57533

Table 5. Goldhill Image Compression Aize for Different Outputs

PSNR (Goldhill image)	4 Outs	6 Outs	8 Outs
24.1494	19938	20506	21095
25.0726	23360	24973	25687
25.2949	24286	26233	27052
26.1554	30093	31712	32777
26.8182	40805	37638	38344
26.8673	43133	37849	38646
27.4289	-	46340	45043
27.9007	-	-	53228
28.076	-	-	58189
28.1162	-	-	59641

From the test results as listed in tables 3, 4 and 5, we find that there is small difference in compression sizes between different outputs in the same PSNR, while the high number of outputs increase the dynamic range of PSNR. Therefore, we select the pe-learned autoencoder with 8 outputs to be used in our compression method.

3.3. Training

As a result of the previous experiments, we will use two simple autoencoders, both of them consist of one hidden layer with different outputs (latent) and the input and output layers are 64 elements which equal to (8 * 8) block of pixels as shown in Figure 4 and Figure 5. We decide to evaluate our method using one output for first autoencoder and 8 outputs for the second autoencoder. The first and second autoencoders are used to compress group1 and group2 respectively.

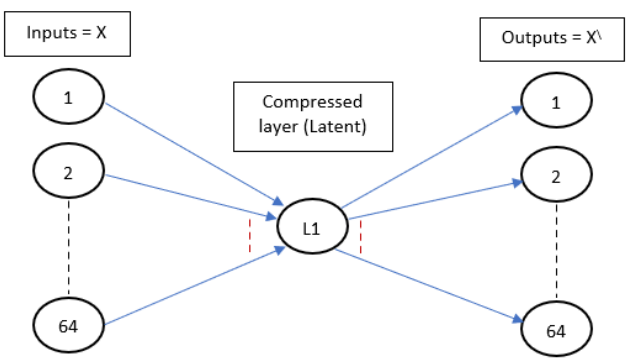

Figure 4. G1 Autoencoder.

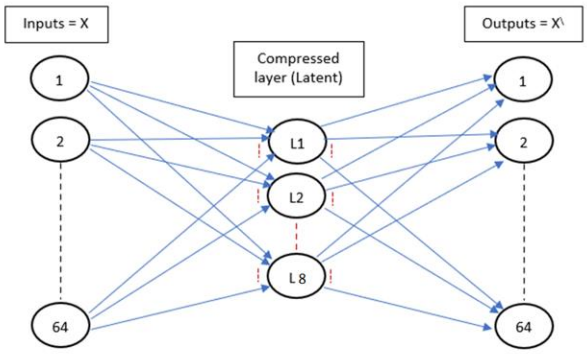

Figure 5. G2 Autoencoder.

Once the architectural designs for the two autoencoders are finalized, we proceed to train these chosen models using the Waterloo image set [34]. Our aim is to acquire the most optimal weights for each autoencoder, the images are segmented into blocks, each consisting of 8×8 pixels. Our training sequence starts with G1 autoencoder, followed by G2. After training we get the most optimal weights for each autoencoder, that make the mean square error as small as possible between the original and reconstructed

4. Evaluation

In order to evaluate the proposed compression method, we will conduct compression on various images with different characteristics. Subsequently, we will utilize three key metrics for the evaluation process as will described in next subsections.

4.1. Evaluation Metrics

Image compression evaluation relies on several metrics to assess the quality and efficiency of compressed images. we will utilize three key metrics for the evaluation process:

1. **Peak Signal-to-Noise Ratio (PSNR):** This metric quantifies the quality of the compressed image compared to the original, it is an expression for the ratio of the maximum possible power of a signal to the power of corrupting noise. Higher PSNR values indicate better quality. The PSNR is usually expressed in terms of the logarithmic decibel scale [35], [36].

$$PSNR = 10 \log \left(\frac{255^2}{MSE}\right) \quad (2)$$

2. **Structural Similarity Index Measure (SSIM):** is a method for measuring the similarity between the compressed and original images by considering luminance (L), contrast (C), and structure (S). It provides a value between -1 and 1, where 1 indicates perfect similarity [35].

$$SSIM(x,y) = \mathrm{L}(x,y).\mathrm{C}(x,y).\mathrm{S}(x,y) \quad (3)$$

$$SSIM(x,y) = \frac{(2\mu_x\mu_y+c_1)(2\sigma_{xy}+c_2)}{(\mu_x^2\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_2)} \quad (4)$$

- x and y are two images being compared.
- μ_x and μ_y are the means of x and y respectively.
- σ_x^2 and σ_y^2 are the variance of x and y respectively.
- σ_{xy} is the covariance between x and y.
- c_1 and c_2 are two constants added for numerical stability.

3. **Bitrate:** Represents the average number of bits used to encode each pixel in the compressed image. There is a relation between the compression ratio (CR) and the bitrate that CR quantifies the degree of compression achieved by comparing the original image size to the size of the compressed image. Higher compression ratios indicate more efficient compression [27], [37].

$$BitRate = \frac{Total\ Encoder\ bits}{No.of\ Pixels} \quad (5)$$

In order to facilitate the evaluation process of our compression algorithm, we develop a LabView program as shown in Figure 6 to preview the images after compression and calculate the evaluation metrics.

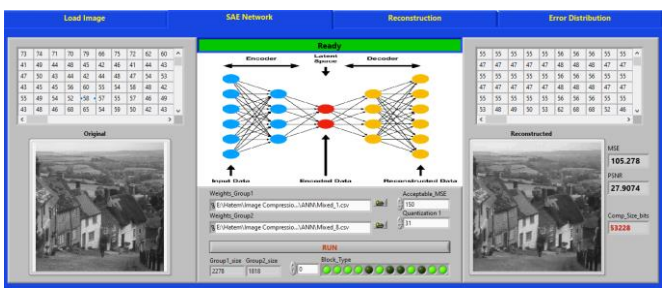

Figure 6. Labview Program.

4.2. Image Compression Results

The assessment for our proposed method occurs by performing our compression method on different images from Waterloo image set, these images are (512 *512) pixels, the compression performed using (8 outputs) autoencoder latent sizes, the Figures 7, 9 and 11 show the relation between PSNR and SSIM values of reconstructed images and the compression Bitrate for our method and JPEG standard. And the Figures 8, 10 and 12 show the visual inspection for our reconstructed images and the JPEG reconstructed images.

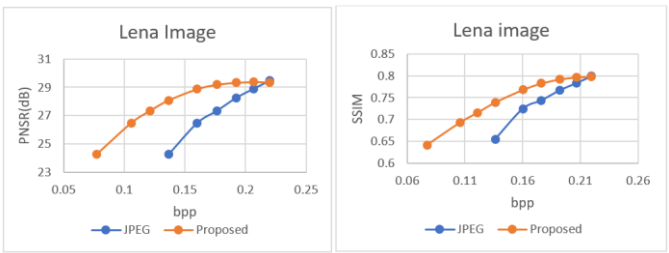

Figure 7. Rate distortion and SSIM for Lena Image

Figure 8. Original and Reconstructed Lena Images from Proposed Method and JPEG

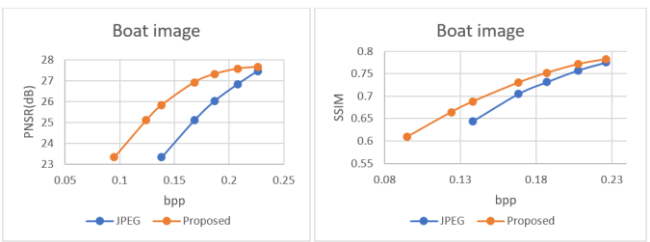

Figure 9. The Rate distortion and SSIM for Boat Image

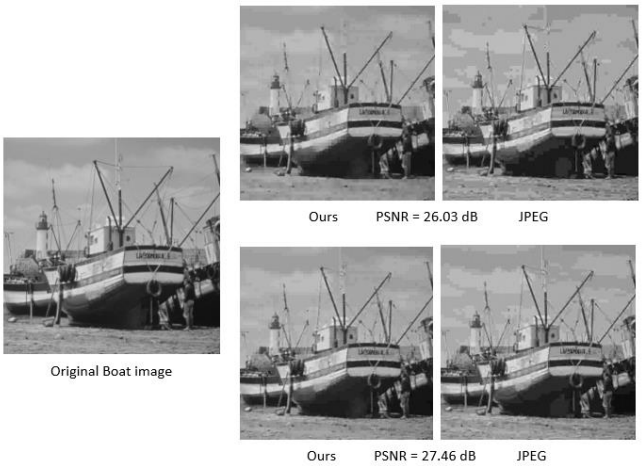

Figure 10. Original and Reconstructed Boat Images from Proposed Method and JPEG

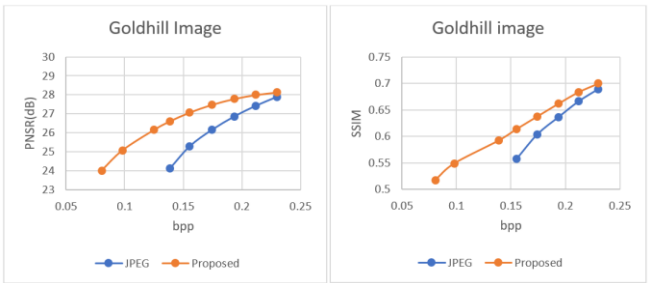

Figure 11. The Rate distortion and SSIM for Goldhill Image

Figure 12. Original and Reconstructed Goldhill Images from Proposed Method and JPEG

From the test results as shown in Figures (7, 9 and 11), we find that, the proposed method has a significant performance in low bit rates less than 0.22 bit per pixel while the proposed method achieves higher PSNR and SSIM than JPEG compression, also the reconstructed images have better visual inspection than JPEG reconstructed image at same PSNR as shown in Figures 8, 10 and 12. The disadvantage of this method is that the PSNR is saturated and cannot achieve high image quality at high bit rates.

5. Conclusion

This work introduces a promising low complexity image compression method, this approach based on two simple pre-learned autoencoders, the image divided into (8 * 8) blocks, the blocks divided into two groups based on the block contents as low details blocks and high details blocks, the first autoencoder which is the simplest one is used to detect the block details and then select the autoencoder which will be used in compression process. From the results we find this method outperforms JPEG significantly in low rates (less than 0.22 bpp) and also it needs low computational complexity. However, the proposed method shows a limited quality that PSNR saturates at bitrate around 0.22 bpp and decreasing the compression effort does not increase the quality. This method seems to be suitable for systems with limited resources that require a low bit rate and low computational complexity, while maintaining an acceptable level of image quality. This research introduces a novel concept that holds potential for future enhancements and refinements to achieve improved results.

References

[1] Rafael C. Gonzalez and Richard E. Woods, “Digital Image Processing”, 4th edition, Personal Education 2018

[2] Khalid Sayood, “introduction to image compression”, 3rd edition, Elsevier 2006.

[3] Gregory K Wallace, "The jpeg still picture compression standard," IEEE transactions on consumer electronics 1992, vol. 38, no. 1, pp. xviii - xxxiv.

[4] Majid Rabbani, "Jpeg2000: Image compression fundamentals, standards and practice," Journal of Electronic Imaging 2002, vol. 11, no. 2, pp. 286.

[5] Y Xie, KL Cheng, Q Chen, “Enhanced Invertible Encoding for Learned Image Compression,” 2021, arXiv:2108.03690.

[6] Trinh Man Hoang, Jinjia Zhou, Yibo Fan;" IMAGE COMPRESSION WITH ENCODER-DECODER MATCHED SEMANTIC SEGMENTATION" Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops 2020, pp. 160-161.

[7] K. Simonyan, and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv 2014, pp. 1409.1556.

[8] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, 2012, pp. 1097–1105.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 770–7785.

[10] Bank D, Koenigstein N, Giryes R. Autoencoders 2020, arXiv preprint arXiv: 2003.05991

[11] Haisheng Fu, Feng Liang, Jie Liang, Yongqiang Wang, he Zhang and Jingning Han, "Fast and High-Performance Learned Image Compression with Improved Checkerboard Context Model, Deformable Residual Module, and Knowledge Distillation" 2023, rXiv:2309.02529v1

[12] Fanqiang Kong, Tongbo Cao, Yunsong Li, Dan Li and Kedi Hu, "Multi-scale spatial-spectral attention network for multispectral image compression based on variational autoencoder", Signal Processing 2022.

[13] Hieu Le, Hernan Santos and Jian Tao, "Hierarchical Autoencoder-based Lossy Compression for Large-scale High-resolution Scientific Data" 2023, arXiv:2307.04216v1.

[14] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto, "Deep Convolutional AutoEncoder-based Lossy Image Compression", arXiv: 1804.09535v1 [cs.CV] 25 Apr 2018.

[15] Wei-Cheng Lee and Hsuch-Ming Hang, "A Hybrid Image Codec with Learned Residual Coding", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops 2020.

[16] Y Xie, KL Cheng, Q Chen, "Enhanced Invertible Encoding for Learned Image Compression," 2021 arXiv:2108.03690.

[17] HiroaKi Akutsu, Akifumi Suzuki, Zhisheng Zhong, Kiyoharu Aizawa;" Ultra Low Bitrate Learned Image Compression by Selective Detail Decoding" Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops 2020, pp. 118-119.

[18] Wong and Bishop, "A Flexible Content-Based Approach to Adaptive Image Compression" ICME 2006

[19] Zhou L, Cai C, Gao Y, Su S, Wu J. Variational autoencoder for low bit-rate image compression. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops 2018, p. 2617–20.

[20] Chen T, Liu H, Ma Z, Shen Q, Cao X, Wang Y. End-to-end learnt image compression via non-local attention optimization and improved context modeling. IEEE Trans Image Process 2021; 30:3179–91

[21] Duan Z, Lu M, Ma Z, Zhu F. Lossy image compression with quantized hierarchical VAEs. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. 2023, p. 198–207.

[22] Gregor K, Besse F, Jimenez Rezende D, Danihelka I, Wierstra D. Towards conceptual compression. Adv Neural Inf Process Syst 2016, 29.

[23] O. Ferraz, G. Falcao and V. Silva, "Gbit/s Throughput Under 6.3-W Lossless Hyperspectral Image Compression on Parallel Embedded Devices," in IEEE Embedded Systems Letters, March 2021, vol. 13, no. 1, pp. 13-16.

[24] Wei Zhao, Zuchen Jia, Xiaosong Wei, d Hai Wang, "An FPGA Implementation of a Convolutional Auto-Encoder" *Appl. Sci.* **2018**, *8*(4), 504.

[25] Zhao W, Jia Z, Wei X, Wang H. "An FPGA Implementation of a Convolutional Auto-Encoder". Applied Sciences 2018.

[26] Salam Fraihat and Mohammed Azmi Al-Betar, "A novel lossy image compression algorithm using multi-models stacked AutoEncoders", Array 2023

[27] Emilien Dupont, Adam Goli´nski, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet, \Coin: Compression with implicit neural representations," arXiv preprint 2021, arXiv:2103.03123.

[28] Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Goli´nski, Yee Whye Teh, and Arnaud Doucet, \Coin++: Data agnostic neural compression," arXiv preprint 2022, arXiv:2201.12904.

[29] Bharath Bhushan Damodaran, Muhammet Balcilar, Franck Galpin, and Pierre Hellier, "RQAT-INR: Improved Implicit Neural Image Compression" 2023, arXiv:2303.03028v1.

[30] Lior Rokach, OdedMaimon and Erez Shmueli, "Machine Learning for Data Science Handbook" (3rd ed.). Springer Nature Switzerland AG 2023, ISBN 978-3-031-24628-9.

[31] Younggrok Song, Sangwon Hyun and Yun-Gyung Cheong, "Analysis of Autoencoders for Network Intrusion Detection", Sensors, 2021.

[32] Trinh Man Hoang, Jinjia Zhou, Yibo Fan;" IMAGE COMPRESSION WITH ENCODER-DECODER MATCHED SEMANTIC SEGMENTATION" Proceedings of the

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops 2020, pp. 160-161.

[33] H. Kubra Cilingir, Sivaramakrishnan Sankarapandian, M. Ozan Tezcan, “Image Compression Using Deep Learning”, Deep Learning Spring 2017, Project Report.

[34] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016, pp. 770–7785.

[35] “Repository.” https://links.uwaterloo.ca/Repostory.html (accessed Jun. 06, 2023)

[36] https://www.ni.com/en-lb/shop/data-acquisition-and-control/add-ons-for-data-acquisition-and-control/what-is-vision-development-module/peak-signal-to-noise-ratio-as-an-image-quality-metric.html

[37] Setiadi DRIM. PSNR vs SSIM: imperceptibility quality assessment for image steganography. Multimedia Tools Appl 2021, 80(6):8423–44.

[38] Charles Poynton , "Digital Video and HD: Algorithms and Interfaces" 2012, (2nd ed.). Morgan Kaufmann Publishers. ISBN 9780123919267.