International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Image Compression of Handwritten Devanagari Text Documents Using a Convolutional Autoencoder

Ambadas B. Shinde*[1], Jayashri Bagade[2], Ratnmala Bhimanpallewar[3], Yogesh H. Dandawate[4]

Submitted: 26/04/2023 **Revised**: 27/06/2023 **Accepted**: 07/07/2023

Abstract: A large number of books/documents written in so many languages are available in digital libraries. These scanned copies of books/documents consume more space on hard disk. If the existing image compression technique like JPEG is used for compressing the textual images then significant amount of information could be lost. Hence we proposed and implemented the Convolutional Autoencoder (CAE) for compression and decompression of handwritten Devanagari document images. As long as compression of document images is concerned, relatively large work was reported for documents having printed text. Comparatively less significant work was done on handwritten Devanagari document images. Convolutional autoencoders have been shown to outperform traditional compression methods like JPEG in terms of compression efficiency and image quality, but requires more computational resources. Convolutional autoencoders have the potential to become a widely adopted approach for image compression in the future, as computing power increases and the demand for higher quality compressed images grows. A lot of experimentation was done by the changing the compression ratios. The PSNR and MSSIM parameters are considered to estimate the performance of compression and decompression results. The compression and decompression results achieved with the help of CAE are much encouraging as compared to the conventional image compression.

Keywords: *Autoencoder, Compression, CNN, Convolutional Autoencoder, Document Compression, Image Compression*

1. Introduction

Digital libraries are extremely important in modern education, especially in pandemics and lockdown situations. Generally the documents are scanned and made available in digital libraries. Storing/preserving all these scanned documents requires a large amount of storage space and considerable time to access these documents. If the size of file is large, it will take longer time to access the file. With the help of document image compression, one can store a large number of books/documents in the available disk space.

Lot of ancient handwritten documents is available in Indian languages. Such historical handwritten documents have lot of importance and significance to preserve the culture and maintain the tradition. Generally the documents are scanned with good quality scanners to prepare their digital copies. Scanning the document images and storing them requires large storage space. To transfer such scanned copies of documents over the internet also has major limitation. The time required to upload and download such scanned documents is significantly high. To ensure prompt transmission over internet, all such documents should be made available in compressed format. Compression technology is available for documents with printed Indian languages [1], [2]. A extremely minute work is found on handwritten text document images. The compression standards are not available for Indian language so far.

Image compression is the technique of minimizing the data which is essential to represent an image while maintaining visual quality. [3]. The purpose of compressing an image is to lessen the size of an image file so that it takes up less storage space and can be transmitted more quickly over the internet. Data compression is one of most promising research domain in field of image processing [1]. There are several popular image compression algorithms, such as JPEG, PNG, and GIF. JPEG is a lossy compression algorithm that is widely used for photographic images. JPEG compression method works on the principle of dividing the image into its small constituents of pixels and then applying a DCT transform [4] to all blocks to convert it into a frequency domain representation. During decompression, the compressed image data is decoded and inverse DCT is applied to reconstruct the original image. PNG is a lossless compression technique that is usually used for images with large areas of unvarying color. It works by encoding the image data using a combination of predictive coding and entropy coding. The predictive

1Electronics and Telecommunication, Vishwakarma Institute of Information Technology, Pune, India
ORCID ID: 0000-0003-0292-9000
2Information Technology, Vishwakarma Institute of Information Technology, Pune, India
ORCID ID: 0000-0002-9709-5530
3Information Technology, Vishwakarma Institute of Information Technology, Pune, India
ORCID ID: 0000-0001-8098-9776
4Electronics and Telecommunication, Vishwakarma Institute of Information Technology, Pune, India
ORCID ID: 0000-0002-2528-6856
** shindesir.pvp@gmail.com*

coding [5] scheme predicts the value of each pixel depending on the value of neighboring pixels, and the resulting prediction error is then encoded using entropy coding, resulting in a compressed image file. GIF is a lossless compression algorithm that is commonly used for animated images. It works by encoding each frame of the animation as a separate image file, and then compressing the sequence of images using Lempel-Ziv-Welch (LZW) compression. The resulting compressed image file contains a sequence of compressed image data for each frame of the animation. Figure 1 shows the stages of lossy image compression.

While image compression can provide numerous benefits, there are also some drawbacks to consider. These include a loss of quality, compression artifacts, compatibility issues, time-consuming compression, limited compression ratios, and compression bias. By considering these factors, it is possible to choose the appropriate compression algorithm for a particular application and optimize the compression settings to achieve the preferred equilibrium between file size and image quality.

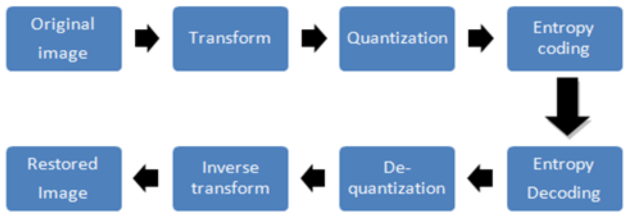


Fig. 1 Lossy image compression

The major issues encountered in conventional document image compression are as below.

Loss of quality: Lossy compression algorithms discard some information during compression process, which can result in a loss of image quality.

Compression artifacts: When image compression is applied, it can result in compression artifacts such as blocking, blurring, or ringing around the edges. These artifacts can be visible and may degrade the visual quality of the image.

Compatibility issues: Different compression algorithms may not be compatible with all types of devices and software.

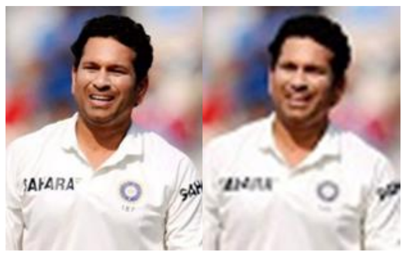
Time-consuming compression: Compression algorithms can be computationally intensive, especially for large image files. This can result in longer processing times, which can be a problem in time-sensitive applications.

Limited compression ratios: While compression algorithms can achieve high compression ratios, there is a limit to how much an image can be compressed without a significant loss of quality. Beyond a certain point, additional compression can result in significant loss of detail and clarity.

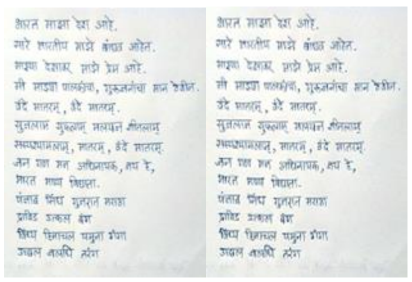
Compression bias: Compression algorithms may have a bias towards certain types of images, resulting in lower compression ratios for some images compared to others.

Table 1 Comparison of image compression

Particulars	*Original Graphics Image*	*Compressed Graphics Image*	*Original Handwritten Text Document Image*	*Compressed Handwritten Text Document Image*
Pixels	591 * 733	60 * 74	1581 * 1761	159 * 177
Size in KB	83.2 KB	3.52 KB	574 KB	15.8 KB
Image quality	Clear	Blur	Clear	Blur
Readability	Person is easily recognizable	Person is recognizable	Cleary Readable	Readability is lost
	Compression Ratio = 97.57		Compression Ratio = 98.93	



(a) (b)



भारत माझा देश आहे.
सारे भारतीय माझे बांधव आहेत.
माझ्या देशावर माझे प्रेम आहे.
मी माझ्या पालकांचा, गुरुजनांचा मान ठेवीन.
वंदे मातरम्, वंदे मातरम्.
सुजलाम् सुफलाम् मलयज शीतलाम्
शस्यश्यामलाम्, मातरम्, वंदे मातरम्.
जन गण मन अधिनायक, जय हे,
भारत भाग्य विधाता.
पंजाब सिंध गुजरात मराठा
द्राविड उत्कल बंग
विंध्य हिमाचल यमुना गंगा
उच्छल जलधि तरंग

भारत माझा देश आहे.
सारे भारतीय माझे बांधव आहेत.
माझ्या देशावर माझे प्रेम आहे.
मी माझ्या पालकांचा, गुरुजनांचा मान ठेवीन.
वंदे मातरम्, वंदे मातरम्.
सुजलाम् सुफलाम् मलयज शीतलाम्
शस्यश्यामलाम्, मातरम्, वंदे मातरम्.
जन गण मन अधिनायक, जय हे,
भारत भाग्य विधाता.
पंजाब सिंध गुजरात मराठा
द्राविड उत्कल बंग
विंध्य हिमाचल यमुना गंगा
उच्छल जलधि तरंग

(c) (d)

Fig. 2 Image compression examples

(a) Original photo image, (b) Compressed photo image (c) Original handwritten text document, (d) Compressed handwritten text document

Figure 2(a) represents the original image and figure 2(b) represents the corresponding compressed image with compression ratio of 97.57. After compressing the image some information is lost but the person in the image is easily recognizable. Similarly figure 2(c) represents the original handwritten Devanagari text document image and figure 2(d) shows the corresponding compressed image with 98.93 compression ratio. From figure 2(d) one can conclude that, if handwritten text document image is compressed then the document quality degrades and its readability reduces.

The details of the images shown in figure 2 before and after compression are summarized in table 1.

The review of existing work was taken in section II. The concept and architecture of the convolutional autoencoders are discussed in section III. Various stages and CAE development of proposed work is discussed in section IV. Section V deals with the investigational results, observations and performance analysis followed by conclusion and references.

2. Literature Review

For many years, the elemental and important study area of image processing has been image compression. The fixed transform matrices, Wavelet Transform (WT) and Discrete Cosine Transform (DCT) along with quantization and entropy coder, are used by conventional image compression algorithms like JPEG [6] and JPEG2000 [7], to compress the picture images. When employed JPEG and JPEG2000 compression standards on handwritten document images, it reduces the visibility of the handwritten documents. This is because touching high frequency components may result in blurring of the image's lines and strokes when scanning documents at various dpi [8]. Chinese-language scanned receipts and handwritten text images were effectively compressed by Danhua and Xudong [8]. This strategy is based on the dissimilarity between an image's foreground and background. With the RLE compression technique, groups of subimages are compressed. The compression of machine printed text images available in Indian scripts using soft pattern matching was presented by U. Garain et al. [9]. The majority of text documents that are printed or written are compressed using Soft Pattern Matching (SPM) and Pattern Matching with Substitution (PMS). These two techniques are more appropriate for printed text as all printed text has a consistent pattern. I. Witeten et al.'s [10] described soft pattern matching-based document compression on printed textual pictures.

Little work has been reported on compressing handwritten text documents having Devanagari script. Since many manually written documents are used officially, so safeguarding of such documents becomes essential. Saving all these documents has high storage space requirements and is difficult to access due to the larger file size. Due the bigger the size of file, the longer is the transfer time. Compressing the document images plays an important role. Deep learning-based image compression techniques like Generative Adversarial Networks (GANs) [11], [12], and Variational Autoencoders (VAEs) [13], [14] have grown in prominence recently. In these techniques, neural networks are used to train a compressed representation of the input picture, which is later decoded to recreate the original image. These techniques are perfect for situations where storage and bandwidth are constrained since they can achieve large compression ratios while preserving good visual quality.

Autoencoder are utilized for image de-noising. In the de-noising task, the autoencoder is considered as a non-linear function that can take out the effect of clamors in the picture [15]. Autoencoder can be considered as better alternative to JPEG image compression.

Particular applications of the autoencoder include dimensionality lessening, creating compacted representations of pictures, and learning generative models [16]. The ability of autoencoders to pull out more compressed codes from pictures while minimizing loss function enables them to outperform current image compression standards in terms of compression performance. As a result, it is anticipated that image compression using deep learning would be more effective and widespread. A lossy image compression architecture based on Convolutional autoencoders (CAE) was suggested by Zhengxue Cheng et al. [17] in their paper. They used an approximation of the rate-distortion loss function to optimize the CAE. They suggested a principle components analysis (PCA) based replacement to produce more zeros in the feature maps in order to provide a more energy-compact representation. The image data is then further compressed using the quantization and entropy coder.

Thierry Dumas et al. [18] proposed a unique transform using autoencoders with variable step sizes during testing. They presented that the learned transformation outperforms other image compression algorithms. Nick Johnston et al. [19] in their research presented an technique for minimizing the computational complexity using automatic network optimization in neural image compression. They found that their method reduces the decoder run-time by more than 50% for neural architecture. Yash Raut et al. [20] in their paper proposed a Convolutional Autoencoder for image compression. They took MNIST dataset having size 28 x 28 RGB images and did up sampling and down sampling of an image for compressing and decompressing the images. Pinar Akyazi and Touradj Ebrahimi [21] in their paper, presented a learning based image compression technique using wavelet decomposition. They trained CAE end-to-end to keep the bit rate less than 0.15 bits per pixel.

3. Convolutional Autoencoders

The Autoencoder is a neural network that tries to replicate the input to the output upon training. It contains a hidden/secret layer h on the inside that explains the code used to represent the input. This network has two components: a decoder that creates a reconstruction, r = g(h), and an encoder function, h = f(x). Figure 3 displays this design. Typically, they are constrained to replicate only roughly and to copy input that closely mimics the training data. The model frequently discovers beneficial characteristics of the data since it must decide which parts of the input should be duplicated in priority.

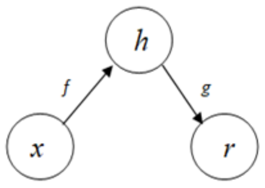


Fig. 3 Autoencoder structure

For many years, the autoencoders are playing crucial role in the development of neural networks. Autoencoders have historically been employed for feature learning or dimensionality reduction. Latent variable models and autoencoders have recently benefited from theoretical concepts. Autoencoders are leading the way in generative modeling. It is possible to think of autoencoders as a specific example of feed forward network and they may be trained using small batch gradients where gradients are calculated by back-propagation method. Recirculation, a learning approach may also be used to train Autoencoders, unlike typical feed forward networks. Recirculation is less frequently utilized in machine learning applications but is seen to be more biologically realistic than back-propagation.

One technique to get helpful characteristics from the autoencoder is to make h smaller than x. Undercomplete autoencoders have code dimensions that are smaller than the input dimension. The autoencoder is forced to capture the most prominent aspects of the training input when learning an incomplete representation. The learning process is merely characterized as lowering a loss function.

$$L(x, g(f(x)))$$

Where, L represents the loss function. Undercomplete autoencoders that has a code measurement less than the input dimension is capable to learn the nearly all important aspects of the data. If the encoder and decoder are provided with many capacities, these autoencoders fail to learn anything valuable. A similar difficulty arises when the dimensions of hidden code and the input layer are same.

Where, L represents the loss function. Undercomplete autoencoders that has a code measurement less than the input dimension is capable to learn the nearly all important aspects of the data. If the encoder and decoder are provided with many capacities, these autoencoders fail to learn anything valuable. A similar difficulty arises when the dimensions of hidden code and the input layer are same.

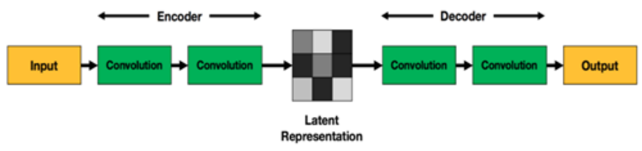


Fig. 4 CAE architecture overview

In this research, we are proposing Convolutional Auto Encoder (CAE) for compression and decompression of handwritten text document images. The CAE compression and decompression architecture is mentioned in figure 4. The CAE accepts the input image. The input image is passed through the multiple convolution layers and max pooling layers connected in cascaded manner. Max pooling layer down samples the input image and compression is achieved. During decompression the compressed image is up sampled and then fed to convolution layer. The final output is the decompressed image.

4. Proposed Methodology

The block diagram of proposed image compression technique using Convolutional autoencoders is referenced in figure 5. This technique has image compression and decompression sections.

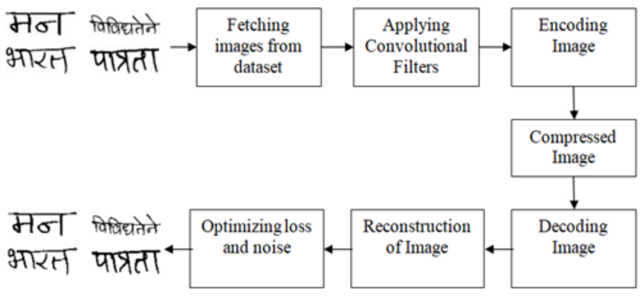

Fetching images from dataset
Applying Convolutional Filters
Encoding Image
Compressed Image
Decoding Image
Reconstruction of Image
Optimizing loss and noise

Fig. 5 Image compression: Proposed Methodology

4.1. Input Images

The considered dataset has 104 images of handwritten Devanagari words. The dataset consists of handwritten Devanagari words of variable length. In this proposed work, we used grayscale images. The size of handwritten Devanagari words is kept uniform to 36 x 96. The used dataset has more than 17,680 images (104 classes with more than 170 images each). The dataset images are stored in the form of 9216 x 1 arrays. The images having size 36 x 96 are resized to 96 x 96. After resizing each word image became ninety-six pixels elevated and ninety-six pixels wide, which gives a total 9216 pixels.

4.2. Convolutional Filters

Extraction of features reduces the data contained in image. We have used Convolutional Autoencoder to carry out the dimensionality reduction. In the proposed technique we applied convolution filters to the image which subsamples the images to progressively smaller images so as to preserve the relative knowledgeable information. The technique of adding a mask to a picture is known as convolution. Consider I(x, y) is the original image and the filter h(x, y) is then the convolution operate f(x, y) can be found with

$$f(x,y) = I(x,y) * \ [elementwise\ ]\, h(x,y)$$

In the first Convolutional layer we have used 128 filters having [3, 3] size with stride [2, 2]. This filter is masked over the image to generate an image map of size 48 x 48 x 128. In each masking operation element-by-element multiplication is performed between the image and filter. This process is continued by shifting the filter to right by the pixel as mentioned by stride. After completing the operation on particular rows the filter is moved to the lower rows by the pixels mentioned in the stride. With such convolution operation the dimension of image reduced to 48 x 48 from 96 x 96. The RELU activation function is used after every convolution operation. RELU is an acronym for rectified linear unit, which exploits the non-saturating function of activation. f(x)=max(0,x). For down sampling, convolution layer is followed by max pooling layer. This structure is followed to reduce the number of pixels in image. After the fifth convolution layer the image is reduced to 12 x 12 x 6 i.e. 864 pixels which signifies that the image is compressed to 10%. The compressed image is represented as latent vector space.

4.3. Image Encoding and Decoding

The encoder performs the compression while decoder does the decompression operation. The encoder reduces the data in an image to present the image in compressed form i.e. reduced size. The compressed image is provided to a decoder. Decoder decompressed the received image to generate the original image.

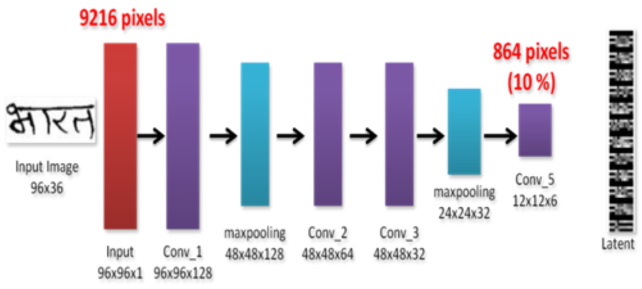


Fig. 6 CAE – Encoder layers architecture

4.4. Compressing the Image

We have used Convolutional layer and CNN's maximum pooling layer in order to reduce the dimensions of image from 96 x 96 x 1 to 12 x 12 x 6. The various layers of the Convolutional autoencoder (image compressor) are shown in figure 6. Five Convolutional layers are followed by the maximum number of pooling levels are used for compressing the handwritten Devanagari text document image.

4.5. Decompressing Image

The image reconstruction is done with help of sequence of symmetrical convolutional layers and deconvolutional layers. In image compression part we have seen that the convolution layer is used as image features extractor which encodes the essential portion of the image and removes the redundant information. The deconvolutional layers reconstruct the picture. The latent space is considered as input for the decompression of image. Up sampling layers and convolutional layers are used to decompress the image. Reverse operations are performed to reconstruct the image. The image decompression method is shown in figure 7.

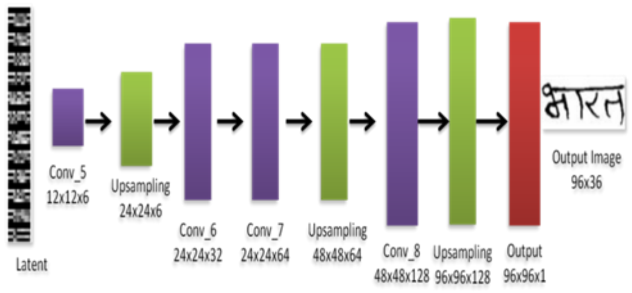


Fig. 7 CAE – Decoder layers architecture

Table 2 shows the details of the convolutional autoencoder designed. As stated earlier the 16 layer CNN model is proposed for the compressing and decompressing the handwritten Devanagari word images. Own dataset is used for experimentation purpose. The dataset has 104 different handwritten words with more than 170 words of each class.

Items	*Details*
CAE model (compression and decompression)	16 layer CNN model for compression and decompression
	1 Input Layer
	9 Convolution layers
	3 Maxpooling layers
	3 Up-sampling layers
Training Data	Dataset of 104 handwritten Devanagari words (17,680 words)
	Training: 70 % & Testing: 30 %
Optimizer	Adam
Batch Size	128
Iterations	200
Quality Check Parameters	PSNR and SSIM
Tool	Google co-lab with GPUs

Table 2 Comparison of image compression

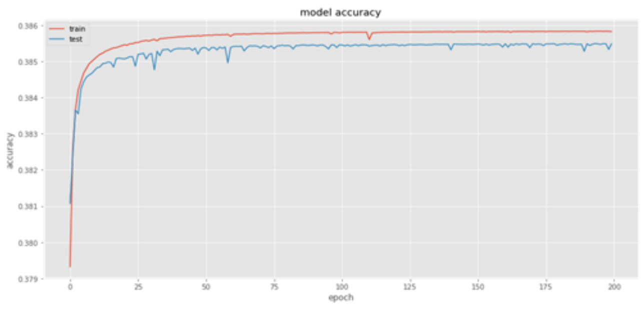
5. Results and Discussion

To compress the handwritten text document image we have proposed Convolutional Autoencoder implemented using CNN architecture. The input image of size 36 x 96 is accepted as input. To maintain the symmetry the image is resized to 96 x 96. This resized image is given to convolution layer. For down sampling, convolution layer is followed by maxpooling layer. This structure is followed to reduce the number of pixels in image. Finally the 96 x 96 grayscale image (9216 pixels) is converted to 12 x 12 x 6 latent space (864 pixels) i.e. original image is compressed to 10 %. The layer structure of CAE for image compression is shown in figure 6.

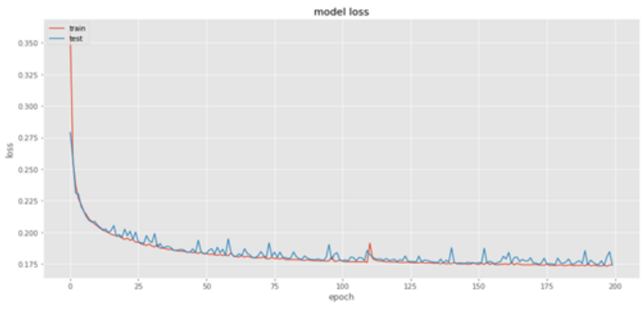
To decompress the image, convolution layer is followed by upsampling layers. Finally the image represented by latent space is reconstructed to original size 96 x 96. The layers of decompression are architecture is shown in figure 7. The 16 layer CAE network is designed and trained for various conditions like number of layers, number of filters, size of filters, learning rate, padding, etc. To maintain the symmetry of CAE network input image initially converted to 96 x 96 x 1 and given to the convolution layers one by one. The middle convolution layer decides the compression ratio.

We made an attempt to compress the handwritten Devanagari word images to various compression ratios. The sufficient experimentation has been done to compress the words images to 25 %, 20 %, 15 %, 10 % and 5 %. For every compression ratio the Structural Similarity Index (SSIM) and Peak Signal to Noise Ratio (PSNR) is calculated. SSIM is an image quality parameter that takes structural resemblance for high-quality approximation of perceived image quality. The sample model accuracy (training and testing) curve as well as model loss (training & testing) curve is shown in figure 8.

Randomly selected 10 words from the dataset are used to demonstrate the results of compression and decompression. Figure 9 shows the result of handwritten Devanagari words compression and decompression. The sample handwritten words as shown in figure 9(a) are used for compression purpose. These words are used as input for 16 layer convolutional autoencoder model. The input words have size 36 x 96.



(a)



(b)

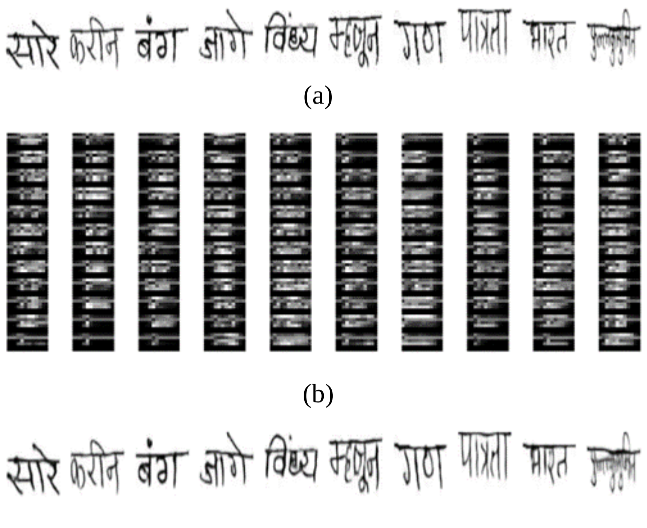
Fig. 8 CAE model accuracy and loss curves

(a) Model training accuracy curve, (b) Model training loss curve

To maintain the symmetry these words are initially resized to 96 x 96 (9216 pixels) and given to convolution layer. The convolution layer is followed by the max-pooling layer. At every convolution layer and max-pooling layer the numbers of pixels are reduced to compress the image. At the last convolution layer the image is converted to 12 x 12 x 6 (824 pixels) to achieve the 10 % compression. The compressed image is now represented as latent space as shown in figure 9 (b).

Table 3. CAE compression result and performance comparison

Compression Ratio	*4*	*4.93*	*6.67*	*10*	*21.33*
CAE Layers	16	16	16	16	16
Input Image Size	96*96*1= 216	96*96*1= 216	96*96*1=9216	96*96*1= 216	96*96*x1= 216
Compressed Image Size	12*12*16=2304	12*12*13=1872	12*12*10 =1440	12*12*6=864	12*12*3=432
Optimizer	Adam	Adam	Adam	Adam	Adam
Epoch	200	200	200	200	200
Learning Rate	0.01	0.01	0.01	0.01	0.01
Batch Size	128	128	128	128	128
Trainable Params	198,625	196,108	193,753	190,865	188,888
PSNR	25.77	25.64	24.42	22.096	16.92
SSIM	0.9667	0.9653	0.9557	0.9350	0.7905



(a)

(b)

(c)

Fig. 9 Result of handwritten word compression and decompression

(a) Sample handwritten Devanagari words, (b) Latent space of the handwritten Devanagari words, (c) Decompressed handwritten Devanagari words

The latent space of the compressed handwritten word is considered as input while decompressing the word. The compressed image is initially fed to the convolution layer. The convolution layer is followed by the up-sampling layers. This combination of the convolution layer and up-sampling layer decompresses the image. The final output of decompression is shown in figure 9. The PSNR and SSIM for 10% compression are 22.59 and 93.59 % respectively.

5.1. Peak Signal to Noise Ratio (PSNR):

The PSNR ratio is the most often used metric for assessing image quality. Here also, the PSNR is used to evaluate the superiority of reconstructed images. The signal is believed to be the original data, while the noise is the inaccuracy caused by compression or distortion.

PSNR is articulated as:

$$PSNR = 10\ log_{10} \frac{(PeakValue^2)}{MSE}$$

Here, Peak Value is the maximal in the image data.

5.2. Structure Similarity Index Method (SSIM)

The SSIM is a perception-based model. This approach views picture deterioration as a change in how structural information is apparent. The phrase "structural information" focuses on the highly interdependent or spatially constrained pixels. The degree of similarity between the restored picture and the original image is calculated by SSIM. The SSIM index technique is calculated based on the computation of three important parameters known as brightness, contrast, and correlation term.

The three words listed below can be used to describe the structural similarity index method:

SSIM $(x, y) = [l(x, y)]^{\alpha} \cdot [c(x, y)]^{\beta} \cdot [s(x, y)]^{\gamma}$

Here, *l* is the luminance of two images,

c is the contrast of two images

s is the structure between two images and

α, β and *γ* are the constants that are positive..

From the similarity index we can conclude that the original image and decompressed image is 93.59 % similar at compression ratio 10. It clearly specifies that compression ratio 10 can reasonably reduces the size of image and decompressed image quality is good.

5.3. Performance Analysis

To estimate the performance of convolutional autoencoder the following parameters were considered: Compression ratio, PSNR and SSIM.

The image is compressed to different compression ratios and again decompressed to check the quality of image. The results of image decompression and performance parameters are mentioned in table 3.

Through the results it can be clearly seen that, the handwritten text document can be compressed to various ratios and again decompressed back to normal. The MSSIM of 10 % compression is approximately 93.6 % and retrieved text is readable.

6. Conclusion

Compressing the document pictures/images using existing approaches/techniques, the clarity (readability) may suffer. The suggested CAE architecture is made up of 16 layers. The photos of handwritten documents were filtered through many processes. The compressed document pictures were fed into the decoder, and the loss and error were optimized using Adam. Based on the experiments and findings, we may conclude that CAE network outperforms the standard encoding approaches such as DCT. SSIM score at 10 % compression is 93.6 % as well as the retrieved text image is readable.

Author contributions

Ambadas Shinde: Dataset generation, pre-processing of data, Methodology, Software, Writing-Original draft preparation.

Jayashri Bagade: Investigation & Validation, Reviewing and Editing

Ratnmala Bhimanpallewar: Investigation & Validation,

Yogesh Dandawate: Conceptualization, Investigation & Validation, Writing-Reviewing and Editing.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] S. V. Khangar, D. P. Bharne, and L. G. Malik, “Compression Strategy for Handwritten Gray Level Document Images,” *Proc. - 2012 Int. Conf. Commun. Inf. Comput. Technol. ICCICT* 2012, pp. 1–5, 2012.

[2] P. S. Metkar and S. S. Thakare, “A Review of Scanned Handwritten Document Compression in Devnagari script,” pp. 457–459, 2018.

[3] K. Satone, A. Deshmukh, and P. Ulhe, “A Review of Image Compression Techniques,” *Proc. Int. Conf. Electron. Commun. Aerosp. Technol. ICECA* 2017, vol. 2017-Janua, no. 1, pp. 97–101, 2017.

[4] W. Xiao, N. Wan, A. Hong, and X. Chen, “A Fast JPEG Image Compression Algorithm Based on DCT,” *Proc. - 2020 IEEE Int. Conf. Smart Cloud, SmartCloud* 2020, pp. 106–110, 2020.

[5] H. T. Sadeeq, T. H. Hameed, A. S. Abdi, and A. N. Abdulfatah, “Image Compression Using Neural Networks: A Review,” *Int. J. online Biomed. Eng.*, vol. 17, no. 14, pp. 135–153, 2021.

[6] G. K. Wallace, “The JPEG Still Picture Compression Standard” pp. 1–17, 1991.

[7] M. Rabbani and R. Joshi, An Overview of the JPEG 2000 Still Image Compression Standard, vol. 17. 2002.

[8] D. Xu and X. Bao, “High Efficient Compression Strategy For Scanned Receipts And Handwritten Documents,” 2009 *1st Int. Conf. Inf. Sci. Eng. ICISE* 2009, pp. 1270–1273, 2009.

[9] U. Garain, A. Mandal, S. Debnath, and B. B. Chaudhuri, “Compression of Scan-Digitized Indian Language Printed Text: A Soft Pattern Matching Technique,” *Proc. 2003 ACM Symp. Doc. Eng.*, pp. 185–192, 2003.

[10] I. H. Witten, T. C. Bell, H. Emberson, S. Inglis, and A. Moffat, “Textual Image Compression: Two-Stage Lossy/Lossless Encoding of Textual Images,” *Proc. IEEE*, vol. 82, no. 6, pp. 878–888, 1994.

[11] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool, “Generative Adversarial Networks for Extreme Learned Image Compression,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-Octob, pp. 221–231, 2019.

[12] Y. Sun, J. Chen, Q. Liu, and G. Liu, “Learning Image Compressed Sensing with Sub-Pixel Convolutional Generative Adversarial Network,” *Pattern Recognit.*, vol. 98, p. 107051, 2020.

[13] V. A. de Oliveira et al., “Reduced-Complexity End-to-End Variational Autoencoder for on Board Satellite Image Compression,” *Remote Sens.*, vol. 13, no. 3, pp. 1–27, 2021.

[14] S. Wen, J. Zhou, A. Nakagawa, K. Kazui, and Z. Tan, “Variational Autoencoder Based Image Compression with Pyramidal Features and Context Entropy Model,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2019-June, pp. 4321–4324, 2019.

[15] Y. Zhang, “A Better Autoencoder for Image : Convolutional Autoencoder,” pp. 1–7.

[16] P. Vincent and H. Larochelle, “Extracting and Composing Robust Features with Denoising.pdf,” pp. 1096–1103, 2008.

[17] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, “Deep Convolutional AutoEncoder-based Lossy Image Compression,” 2018 *Pict. Coding Symp.*, pp. 253–257, 2018.

[18] T. Dumas, C. Guillemot, and I. R. Bretagne-atlantique, “Autoencoder Based Image Compression : Can the Learning be Quantization Independent,” 2018 *IEEE Int. Conf. Acoust. Speech Signal Process.*, no. 1, pp. 1188–1192, 2018.

[19] N. Johnston, A. Gordon, and J. Ballé, “Computationally Efficient Neural Image Compression.”

[20] Y. Raut, T. Tiwari, P. Pande, and P. Thakar, “Image Compression Using Convolutional Autoencoder,” *Lect. Notes Electr. Eng.*, vol. 601, pp. 221–230, 2020.

[21] P. Akyazi and T. Ebrahimi, “Learning-Based Image Compression using Convolutional Autoencoder and Wavelet Decomposition,” 2000.

[22] Mondal, D., & Patil, S. S. (2022). EEG Signal Classification with Machine Learning model using PCA feature selection with Modified Hilbert transformation for Brain-Computer Interface Application. Machine Learning Applications in Engineering Education and Management, 2(1), 11–19. Retrieved from http://yashikajournals.com/index.php/mlaeem/article/view/20

[23] Anand, R., Ahamad, S., Veeraiah, V., Janardan, S. K., Dhabliya, D., Sindhwani, N., & Gupta, A. (2023). Optimizing 6G Wireless Network Security for Effective Communication. In Innovative Smart Materials Used in Wireless Communication Technology (pp. 1–20). IGI Global.

[24] Mohan, D. ., & Nair, L. R. . (2023). A Robust Deep Model for Improved Categorization of Legal Documents for Predictive Analytics . International Journal on Recent and Innovation Trends in Computing and Communication, 11(3s), 175–183. https://doi.org/10.17762/ijritcc.v11i3s.6179

[25] Pande, S. D., Kanna, R. K., & Qureshi, I. (2022). Natural Language Processing Based on Name Entity With N-Gram Classifier Machine Learning Process Through GE-Based Hidden Markov Model. Machine Learning Applications in Engineering Education and Management, 2(1), 30–39. Retrieved from http://yashikajournals.com/index.php/mlaeem/article/view/22