Fully Convolutional Auto-encoder Based Multi-Stage Encoding for Image Compression

Mr. Giridhar S. Sudi,
Asst. Professor
KLS GIT, Belagavi

Dr. Mahendra M. Dixit
Professor
KLS VDIT, Haliyal

Abstract

In this paper, we propose to perform "Image Compression Using the Framework of Fully Convolutional Auto-encoder Based Multi-Stage Encoding". Image that contains co-related data can be compressed easily. We also propose to increase the correlation of the image by smoothening the edges by convolving the image with a known PSF. The usage of fully convolutional Auto-encoder helps us to further compress the encoded data using DCT and other encoding techniques. We propose to train the system and demonstrate the results on the COCO dataset. The image compression done using the proposed framework is ideally expected to be lossless. But, the trained Auto-encoder model is expected to show some inaccuracy in terms of efficient reconstruction. Hence, the total loss in the compression technique is attributed to the inaccuracy of the training. It is expected that the proposed system compresses the image with a larger compression ratio in comparison with the state-of-the-art JPEG compression.

Index terms: Image compression, auto-encoder, convolution, PSF, DCT, image reconstruction

I. Introduction

Image compression is a type of data compression that can be applied to digital images. It reduces the cost for storage and transmission of the images. The image compression algorithms generally take advantage of the visual perception and the statistical properties of image data to provide superior results compared with generic data compression methods which are used for other digital data. Image compression can be of two types 1) Lossy and 2) Lossless.

In lossless type of image compression we expect the original image and the reconstructed image to have same quality. i.e. we expect it not to change. In lossy mode of compression, we focus on visual perception rather than the exact recovery. Lossy compression technique always has larger compression rate when compared to lossless compression.

We find many image compression techniques in the literature. The algorithms available in the literature can be classified into five major categories.

1. Wavelet based compression techniques
2. JPEG/DCT based techniques
3. Vector quantization based techniques.
4. Fractal based methods
5. Generic algorithm

Wavelet based methods are very popular in applications where the compression ratio is needed to be of around 0.5 bits per pixel. The chief advantages of this technique include the high compression ratio, low encoding complexity and it provides no blocking effects. However it comes up with the disadvantages in coefficient quantization, bit allocation. It is seen that it is not so efficient on all type of images.

The JPEG/DCT based method is the current standard in image compression. It provides high quality and smaller degree of compression. It is also comparatively faster when compared with the other state of the art algorithms. However, it too faces difficulties of bit allocation and coefficient quantization.

The VQ based approach is simple and has no coefficient quantization. However, this approach is little slow in terms of its codebook generation and it offers small bit per pixels.

The fractal based methods comes up with a solid mathematical encoding framework. It is generic to any resolution. However it is observed that this technique is a little slow.

The generic algorithms are generally capable of handling complexity and irregular solution spaces and they are also capable. But, the fitness function evaluation takes lot of time and hence, this technique is generally not so efficient on all the cases.

All the above proposed methods are designed for generic data. i.e. these techniques can be used to compress any given image. However, this need not be true in any real time scenario. For real time applications, the image space used in an application is limited. Hence, the algorithm that can provide efficient image compression for that image space is generally considered instead of the algorithm that can do image compression on the entire natural image space, but with a bad image compression ratio.

The quench to develop such an algorithm is generally solved by the usage of the deep neural networks. An encoder decoder based network can be trained using the concept of auto-encoder in deep learning on a specific dataset to provide high image compression. But, the deep learning techniques learn some unknown transformation and forcing it to learn a transformation (encoding) that is better than the state of the art DCT would require huge data. The scarcity of the data required for training pose many problems in the usage of deep neural networks.

In this paper, we propose to address the issue of sparse dataset to achieve better image compression by combining the neural network based compression and the DCT based compression technique. We also propose to exploit the data compaction property of the DCT algorithm which states that the image with high correlation can have better data compaction.

Towards this, we make the following contributions:

1) We propose a novel deep learning and DCT transform based image compression algorithm a. We propose a fully convolutional auto-encoder with max-pool skip connections for efficient learning.

b. We train the proposed neural network using the self-generated handwritten digit data.

c. We perform channel wise DCT transform on the encoded data that is obtained from the auto- encoder.

2) We propose to increase the correlation in the input image by blurring it with a known PSF. To nullify the effect of blurring, we propose to de-blur the output image by performing sharpening and morphological operations on it.

3) We demonstrate the results on the handwritten digit images and compare the same with the state of the art JPEG compression and show that the proposed algorithm overpowers the state of the art JPEG in all aspects.

The proposed system

The proposed system has three phases

1. Training Phase
2. Compression Phase
3. Reconstruction Phase

1) **Training Phase:** In this phase, we construct a "fully convolutional auto-encoder network" and learn the weights of the encoder and decoder neural networks. The correlation information can be increased in the neural network by applying motion blur of specific kernel. Blurred image is used to learn the weights. We propose to use skip connections for the max-pool layers to achieve better learning.

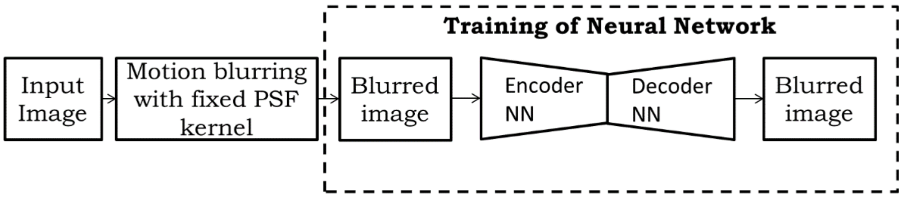


Fig 1. The training phase

2) **Compression Phase:** In this phase, we propose to compress the image with high compression ratio. We first increase the correlation in the data using the technique of motion blurring. Then the blurred image is passed to the encoder block to get the encoded image (E1). As the network is fully convolutional in nature, we get a two dimensional matrix as output from the encoder neural network. This 2D matrix can now be sent to DCT and then we can perform run length coding and Huffman coding to achieve higher compression ratio.

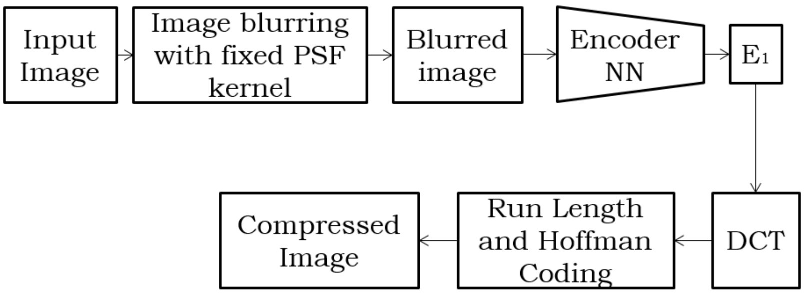


Fig 2. The compression phase

3) Reconstruction Phase: In this phase, we first perform decoding of run length and Huffman coding and then perform IDCT to obtain the 2D matrix (E1) that was generated by the encoder NN. Now, E1 is sent to the decoder network to get back the blurred image. This image is now de-convolved with the same PSF function used for blurring. This gives us the input image back.

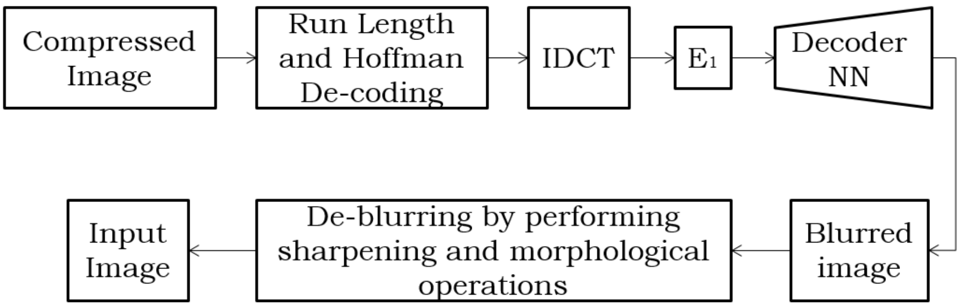


Fig 3. The reconstruction phase

For the purpose of the assignment, we do not implement or demonstrate the results of the Huffman coding or the run length coding. This is because run length coding and Huffman coding perform reduction in number of bits per symbol. We propose to show massive reduction in the number of symbols retained after compression and hence it is evident that the proposed system overpowers the state of the art algorithms even after the run length coding and Huffman coding is done.

The proposed system specifications

The system specifications can be categorized into two categories:

1. Information on the blur PSF used to blur the images
2. Details of the auto-encoder network.

Information on the blur PSF used to blur the images

In this paper, we propose to use motion blurring on the image. To achieve this, we resort to MATLAB coding by using the inbuilt function

h = fspecial ('motion', len, theta).

For this function, 'len' and 'theta' are the motion blur parameters and are set to be 2 and 5 respectively.

Details of the auto-encoder network.

We propose a new fully convolutional auto-encoder with skip connections for every max-pool layers to achieve better learnability. i.e. we propose to maintain the switch variables that identify the location of the maximum value during encoding (pooling) and the gradient is passed back to the same location during un-pooling. This gives better accuracy. A creative visualization of the same is shown in the figure below.

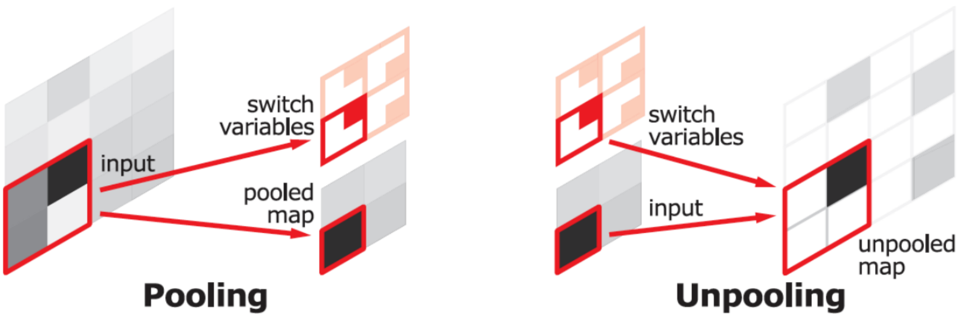


Fig 4. Auto-encoder network

The designed model and its corresponding parameter list can be summarized in the table below:

Table 1. Summary of the parameter list for the designed model

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 6, 124, 124]	456
MaxPool2d-2	[[-1, 6, 62, 62], [-1, 6, 62, 62]]	0
Tanh-3	[-1, 6, 62, 62]	0
Conv2d-4	[-1, 12, 58, 58]	1,812
MaxPool2d-5	[[-1, 12, 29, 29], [-1, 12, 29, 29]]	0
Tanh-6	[-1, 12, 29, 29]	0
Conv2d-7	[-1, 16, 25, 25]	4,816
Tanh-8	[-1, 16, 25, 25]	0
Tanh-9	[-1, 16, 25, 25]	0
Conv2d-10	[-1, 12, 21, 21]	4,812
Tanh-11	[-1, 12, 21, 21]	0
Tanh-12	[-1, 12, 21, 21]	0
Conv2d-13	[-1, 8, 17, 17]	2,408
Tanh-14	[-1, 8, 17, 17]	0
ConvTranspose2d-15	[-1, 12, 21, 21]	2,412
Tanh-16	[-1, 12, 21, 21]	0
ConvTranspose2d-17	[-1, 16, 25, 25]	4,816
Tanh-18	[-1, 16, 25, 25]	0
ConvTranspose2d-19	[-1, 12, 29, 29]	4,812
Tanh-20	[-1, 12, 29, 29]	0
MaxUnpool2d-21	[-1, 12, 58, 58]	0
ConvTranspose2d-22	[-1, 6, 62, 62]	1,806
Tanh-23	[-1, 6, 62, 62]	0
MaxUnpool2d-24	[-1, 6, 124, 124]	0
ConvTranspose2d-25	[-1, 3, 128, 128]	453

```
Total params: 28,603
Trainable params: 28,603
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.19
Forward/backward pass size (MB): 4831.71
Params size (MB): 0.11
Estimated Total Size (MB): 4832.00
----------------------------------------------------------------
```

The training is done on the self-prepared handwritten dataset of 768 images of size 128x128x3. We train the network on MSE loss at a batch size of 16 for 20 epochs. The final loss we observe is at 0.0054.

Results and Discussion

In this section we propose to demonstrate the original image and the image compressed using the proposed algorithm and also the image compressed using the state of the art JPEG algorithm for two datasets. We estimate the compression ratio using the following formula.

$$CR = M/N$$

Where,

M = number of elements in the original image

N = number of elements in the compressed array.

As we are not implementing the Huffman coding and the run length coding, we are not demonstrating the results on the number of bits present in the original and the compressed images.

We also quantitatively demonstrate the similarity between the original image and the reconstructed image using the quantitative parameters such as NMI (normalized mutual information) and SSIM (structural similarity index metric).

Results of dataset 1

The following images show the compression and decompression on the image (a) using the proposed algorithm in (b) and the state of the art JPEG technique in (c). The image is of the size 128x128x3 and hence, the number of elements in the original image M = 49152. The proposed algorithm reduces the number of elements to 6 (N = 6) and the JPEG reduces it to (N = 264)

Therefore, the compression ratios are:

CR (proposed algorithm) = 8192

CR (JPEG algorithm) = 186.18

Thus, the proposed algorithm produces 44 times better compression when compared to JPEG compression

(a) (b) (c)

Fig 5. Results of dataset-1. (a) is the input image, (b) is the image compressed using the proposed algorithm and (c) is the image compressed using the JPEG algorithm

The NMI and SSIM values for the decompressed images are as follows:

Image	NMI	SSIM
Decompressed using the proposed algorithm	0.603492614143437	0.560602761755208
Decompressed using the JPEG algorithm	0.883657259311433	0.641854119237740

Higher NMI and Higher SSIM indicate better reconstruction.

Results of dataset 2

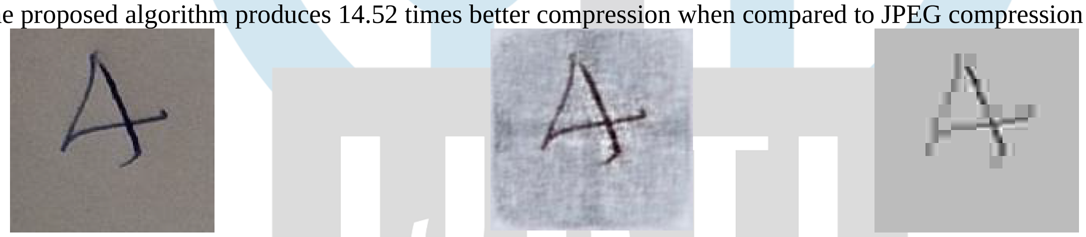
The following images show the compression and decompression on the image (a) using the proposed algorithm in (b) and the state of the art JPEG technique in (c). The image is of the size 128x128x3 and hence, the number of elements in the original image M = 49152. The proposed algorithm reduces the number of elements to 19 (N = 19) and the JPEG reduces it to (N =276)

Therefore, the compression ratios are:

CR (proposed algorithm) = 2586.9

CR (JPEG algorithm) = 178.08

Thus, the proposed algorithm produces 14.52 times better compression when compared to JPEG compression



(a) (b) (c)

Fig 6. Results of dataset-2. (a) is the input image, (b) is the image compressed using the proposed algorithm and (c) is the image compressed using the JPEG algorithm

The NMI and SSIM values for the decompressed images are as follows:

Table 2. Summary of NMI and SSIM results

Image	NMI	SSIM
Decompressed using the proposed algorithm	0.666967522288654	0.607654984201829
Decompressed using the JPEG algorithm	0.796112884806761	0.558222754205304

Higher NMI and Higher SSIM indicate better reconstruction.

Conclusion

In this paper we have proposed a new image compression algorithm that combines the properties of the deep learning and the DCT based techniques. We have proposed to blur the image to increase the data correlation. The system is assessed qualitatively and quantitatively for the compression as well as decompression. We observe that the proposed algorithm performs better than the state-of-the-art DCT based technique.

REFERENCES

[1] E. M. Saad, A. A. Abdelwahab and M. A. Deyab, "Using feed forward multilayer neural network and vector quantization as an image data compression technique," *Computers and Communications, 1998. ISCC '98. Proceedings. Third IEEE Symposium on*, Athens, 1998, pp. 554-558.doi: 10.1109/ISCC.1998.702592

[2] V. D. Raut and S. Dholay, "Analyzing image compression with efficient transforms & multistage vector quantization using radial basis function neural network," *2015 IEEE International Conference on Engineering and Technology (ICETECH)*, Coimbatore, 2015, pp.1-6.doi: 10.1109/ICETECH.2015.7275009

[3] P. Natu, S. Natu and T. Sarode, "Hybrid image compression using VQ on error image," *2017 International Conference on Intelligent Communication and Computational Techniques (ICCT)*, Jaipur, 2017, pp. 173-176.doi: 10.1109/INTELCCT.2017.8324040

[4] W. K. Yeo *et al*., "Grayscale medical image compression using feed-forward neural networks," *2011 IEEE International Conference on Computer Applications a nd Industrial Electronics (ICCAIE)*, Penang, 2011, pp. 633-638.doi: 10.1109/ICCAIE.2011.6162211

[5] P. K. Shah, R. P. Pandey and R. Kumar, "Vector Quantization with codebook and index compression," *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*, Moradabad, 2016, pp. 49-52.doi: 10.1109/SYSMART.2016.7894488

[6] W. Zhang, H. Li and X. Long, "An improved classified vector quantization for medical image," *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, Auckland, 2015, pp. 238-241.doi: 10.1109/ICIEA.2015.7334118

[7] Hussain, A. J., Ali Al-Fayadh, and Naeem Radi. "Image compression techniques: A survey in lossless and lossy algorithms." *Neuro-computing* (2018).

[8] Man, Hong Ren Wu, Sophie Liu, and Xinghuo Yu, *"A New Adaptive Back propagation Algorithm Based on Lyapunov Stability Theory for Neural Networks"*, IEEE transactions on neural networks, Vol. 17, No. 6, November 2006, Zhihong.

[9] Dr. S. Sathappan, "*A Vector Quantization Technique for Image Compression using Modified Fuzzy Possibilistic C-Means with Weighted Mahalanobis Distance*", International Journal of Innovative Research in Computer and Communication Engineering, Vol.1, Issue 1, March 2013.

[10] Rohit Kumar Gangwar, Mukesh Kumar, A. K. Jaiswal, Rohini Saxena, "*Performance analysis of image compression using fuzzy logic algorithm*", Signal & Image Processing: An International Journal (SIPIJ) Vol.5, No.2, April 2014.