International Journal of Innovative Research in Engineering
Volume 3, Issue 3 (May-June 2022), PP: 19-23.
www.theijire.com
ISSN No: 2582-8746

Image Compression Using Deep Auto Encoder

Dr. S. JANA[1], Arjun Sarraf[2], Rahul Kumar[3], Shubham Kumar[4]
[1,2,3,4]*ECE,* Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, *Tamilnadu, India.*

How to cite this paper:
Dr. S. JANA[1], Arjun Sarraf[2], Rahul Kumar[3], Shubham Kumar[4], "Image Compression Using Deep Auto Encoder", IJIRE-V3I03-19-23

Abstract: *The quick Rise of various online Platforms has resulted in a massive influx of data, large in the form of photographes and vidéos. Uncompressed MultiMedia data, such as graphics, audio, and video, requiers a significant amount of Storage space and bandwidth. transmission. The need for appropriate image compression to combat This excessive data transmission is critical. It has become unavoidable to use approches. Image compression is the process of reducing the size of an image in order to Save space. Storage and transmission that Is useful Various approaches have emerged to overcome this challenge, but none of them had been successful. The rebuilt image suffers severe data loss, which is a big disadvantage. To deal with the situation This study promotes the value of deep Learning by establishing a Framework. Convolutional Autoencoder is a type of autoencoder that uses a Convolutional To develop a better picture compression model, a convolutional autoencoder model with 20 different layers and filters was created. This unsupervised machine learning technique compresses images using back propagation and reconstructs the input image with the least amount of loss. A new instance has been added to the architecture to do denoising with the least amount of data loss possible. The architecture has demonstrated its success in image compression and denoising, but It has also cleared the way for additional research into the model's improvement in terms of compression factor and data loss reduction in higher-dimensional images.*

Keywords*- Image compression, Autoencoder, denoising image*

I. INTRODUCTION

The relevance of numerous online forums and social networks has grown as a result of digitalization, with photos and videos serving as the key sources of information. With the escalating making use of the web host there is a pressing demand for image uploads that are faster and have better file quality. There is no significant data loss at this size. The results of extensive computer vision research have been compiled. Deep learning techniques have progressed to the point where they can handle the challenge successfully. Image compression techniques are a source of concern. Techniques for image compression have progressed. Over the decades, as its value has grown. The first picture compression algorithms, such as JPEG, relied heavily on block diagrammed encoders or decoders that required human intervention. Traditional techniques, on the other hand, were constrained to specific image contents and formats because they relied on inflexible redesigned matrices like the wavelet transform and discrete cosine transform, as well as an entropy coder and quantization. Furthermore, classic image compression approaches were challenged by the emergence of high-resolution photos and improved formats. With the introduction of deep learning techniques, image compression can be improved by using an autoencoder and decreasing the loss function. To address the shortcomings of traditional methods, a convolutional autoencoder was developed with the goal of achieving successful image compression with minimal loss in the reconstructed image produced by the decoder, as well as the addition of a new instance to the existing architecture with the goal of denoising. Three experiments were conducted to evaluate the architecture's performance in image compression and denoising, as well as to explain its success, flaws, and future potential.

II. MOTIVATION

In practically every domain, the fast digitization has created a need for optimal storage and transport of information in the form of photos and videos. With the ever-increasing bandwidth generated by information from multimedia and digital representations of images, picture compression has become a need. Uncompressed multimedia data, such as graphics, audio, and video, requires a significant amount of storage space and bandwidth for transmission. Despite rapid increases in mass density storage, the speed of the processor and the digital communicational system has created a demand for greater data storage capacity. This has outstripped the capabilities of current technologies. Furthermore, as the number of data-intensive web-based apps grows, Along with the need for signal encoding and compression technologies, there was a demand for optimum signal compression for storage in the central repository. Image compression has evolved as a solution to these problems, prompting the start of research on the construction of an autoencoder model capable of image compression with the least amount of loss conceivable.

III. RESEARCH OBJECTIVE

1. Image compression

Compression of images the autoencoder can be used to compress images. In this job, the Autoencoders hidden layer must be smaller than the output layer. Backpropagation training using input values that are exactly the same as the target values forces the autoencoder to learn a low-dimensional representation of the input data. The compressed data is what activates the hidden layer. The image encoder is the first portion of this network, while the decoder is the last part.

2. Image de-noising

The process of reducing noise from an image is known as de-noising. Autoencoder can also be used for image de-noising. In the picture de-noising task, we treat the autoencoder as a non-linear function capable of removing the image's effect of noises. The image is fed with random noise to train the network, with the original image sans noise as the output target. This motivates the autoencoder to figure out a way to remove the noise and rebuild the image.

IV.METHODOLOGY

Convolutional Autoencoder (CAE)

The basic structure of the convolutional autoencoder is extended by altering the fully linked layers to convolution layers. The size of the input layer is the same as the output layer in the simple autoencoder, but the decoder network is changed to Convolution layers and the decoder network is changed to transposed convolutional layers. Convolutional Autoencoders are a type of Convolutional Neural Networks that are used to train convolution filters unsupervised. They are commonly used in image reconstruction to learn the best filters and thereby reduce reconstruction mistakes. Once they've been taught on this activity, they can use it to extract features from any input. Unlike general Autoencoders, which ignore the 2D picture structure entirely, convolutional AutoEncoders are general-purpose feature extractors. The image must be unrolled into a single vector in Autoencoders, and the network must be designed with the number of inputs constraint in mind.

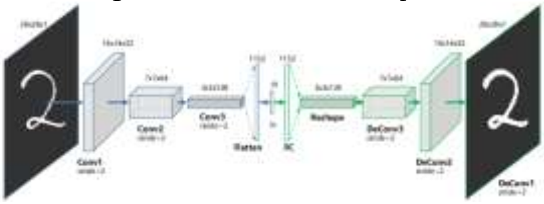

Fig-block diagram of a Convolutional Autoencoder.

Processing

CAE is a deep neural network model that can take in data, compress and grasp its structure over a number of layers, and then create that data again. Works especially well with image data. An autoencoder employs two types of networks. The first is known as an encoder, and the second is known as a decoder. The decoder is merely a reflection of the encoder's layers. The encoder's job is to take two or more dimensional data (for example, an image) and turn it into a single 2D vector that represents the full image. The amount of elements in the 2D vector changes according to the task at hand. It could consist of one or more elements. The fewer elements in the vector, the more difficult it is to precisely reproduce the original image. We compress the input image by encoding it as a vector with a small number of elements. Each image in the MNIST collection, for example, is 28x28 pixels in size. A 2D vector is generated by an encoder from an input image. You can utilize any of the layers offered, such as dense, convolutional, dropout, and so on.

Encoder

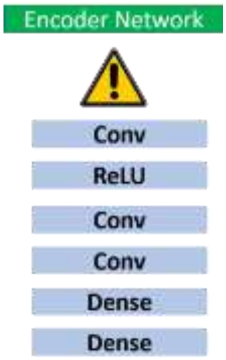

Fig-Encoder Network.

The encoder's last layer generates a 2D vector, which is subsequently given to the decoder. The decoder job is to reconstruct the original image as accurately as possible. The encoder is merely reflected in the decoder. The architecture of the decoder is shown in the next picture, which is based on the encoder design in the preceding figure.

Decoder

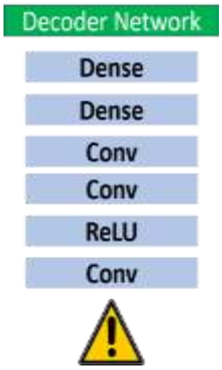

Fig- Decoder Network

The loss is measured by computing the difference between the pixels in the original and reconstructed images. It's worth noting that the decoder's output must be the same size as the source image.

V.RESULT AND DISCUSSION

1. Dataset

Number is the dataset we chose. It has around 10,000 images of various numbers.
Some of them were chosen at random. Because the focus of this study is on numbers, we manually cropped the primary image to eliminate the background noise.

1. Compress the MNIST dataset with stochastic compression and noise.
2. On the dataset, train a compressing and denoising autoencoder.
3. Recover the original digits from the dataset automatically.

We applied compression and noise to the MNIST dataset to demonstrate a compressing and denoising autoencoder in action, dramatically lowering the image quality to the point that any model would struggle to correctly categories the digit in the image. Using our autoencoder for compression and denoising, we were able to recover the original data after removing the noise from the image and compressing it (i.e., the digit).

2. Image Compression

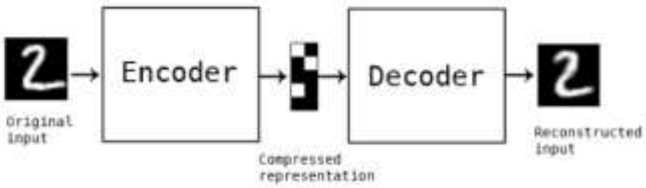

Fig- A compressed image using CAE

For image compression, Autoencoder can be utilized. The AutoEncoders hidden layer must be smaller than the output layer in this challenge. Backpropagation training with input values identical to target values forces the autoencoder to learn a low-dimensional representation of the input data. Compressed data is used to activate the hidden layer. The picture encoder comes before the decoder in this network. To make the results comparable, we manually adjusted the number of units and channels in the hidden layers to make both Autoencoders' compression representations the same dimension.
We increase the hidden layer size to 64x32 for the convolutional autoencoder. We increase the size of the concealed layer to 64x32 pixels. We decided to train CAE 20 epochs. We keep track of each epoch's loss and show the rebuilt image of the most recent epoch. Clearly, CAE's reconstruction result is clearer, demonstrating that CAE can compress data with less information loss even when the number of free parameters is significantly smaller than the number of free parameters. In addition to picture restoration quality, CAE requires less training time to achieve a tolerable loss.

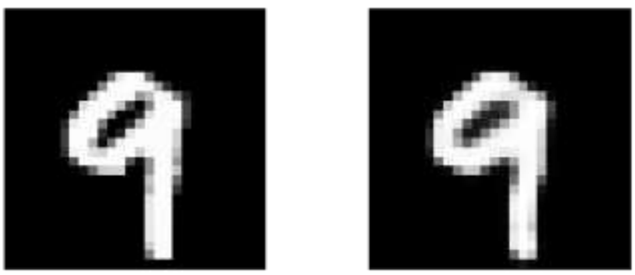

Fig-image compression result of CAE: left is the original image and right is reconstructed image.

3. **Image Denoising**

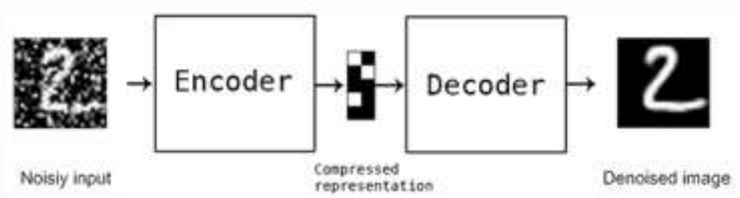

Fig-A denoising convolutional autoencoder processes a noisy image.

De-noising of images another application for autoencoder is image de-noising.
We treat the autoencoder as a non-linear function that can remove the effect of noises in the image in the image de-noising task. We train this network by feeding it an image with random noise , with the original image as the output target. This motivates the autoencoder to develop a function for removing noise and reconstructing the image.

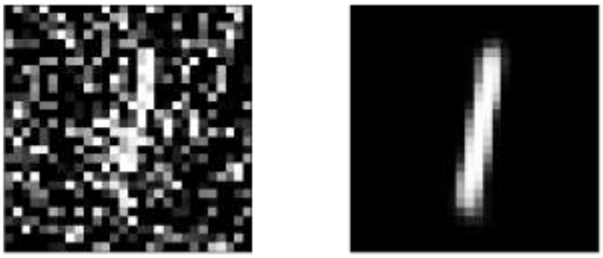

Fig-Denoised result of CAE: left is the noised image and right is de-noised image.

VI.TOOL USED

1. Python
2. Kaggle
3. Jupiter Notebook

VII.CONCLUSION AND FUTURE WORK

From the experiment above, we can observe that utilizing a convolution autoencoder improves both image compression and image de-noising in the experiment above. The following are some possible explanations for the improvement:

1. Because we're working with image data, a convolution layer is a superior choice for capturing spatial information.
2. Instead of utilizing a single hidden layer in the CAE, employ many layers to extract the image's high-level features, which results in a more accurate representation.
3.The number of free parameters in the fully linked layer is more than in the multi convolution layer, making the basic auto encoder difficult to train and time consuming.

The suggested neural network model is trained and tested for three distinct block sizes, each with a different saving percentage. The model with 28x28 blocks performs better on image reconstruction and measurements than models with 14x14 and 7x7pieces. As a result, even with the same saving percentage, the block size is a crucial factor. However, both models with 4x4 and 8x8 blocks perform similarly in human visual perception. In addition, when the model was compared to the JPEG compression technique, the results were extremely similar, and the reconstruction of images from both algorithms appears to be identical in terms of human visual perception. As a result, the proposed model's capacity to handle generalization for a collection of specific data sets has been tested. As a result, the suggested model is capable of managing generalization for a group of specific images, as well as holding large enough training data while remaining small enough for the network's size. In order to increase the performance of the existing compression technique, the proposed model can be implemented utilizing a CAE. The CAE can be used to compress a colour image because of its capacity to handle numerous components of an image. This feature is useful for a variety of investigations because it eliminates the requirement for a complicated modelling strategy. Deep learning is also utilized in picture classification and object recognition, image segmentation, image compression, and other applications, and has resulted in some amazing discoveries and advancements. The deep learning-based image compression approaches have been evaluated and contrasted in this study, which is especially useful for new research on deep learning-based image compression techniques.

IV.ACKNOWLEDGEMENT

I would like to acknowledge my supervisor, Dr. S. JANA for the guidance, support, and Encouragement that she has provided throughout the course module.

REFERENCES

1. Gedeon, T. D., Harris, D.: Progressive image compression. In Neural Networks, 1992. IJCNN., International Joint Conference on (Vol. 4, pp. 403-407). IEEE (1992).
2. Gedeon, T. D., Harris, D.:Network reduction techniques. In Proceedings International Conference on Neural Networks Methodologies and Applications, Vol. 1, pp. 119-126 (1991)
3. Jolli_e, I. T.: Principal component analysis and factor analysis. In Principal component analysis pp. 115- 128. Springer, New York, NY(1986)..
4. Nasrabadi, N. M. : Pattern recognition and machine learning. Journal of electronic imaging, 16(4), 049901 (2007).
5. Dua, D., Karra Taniskidou, E.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine,CA: University of California, School of Information and Computer Science(1986).
6. Sra, S., Dhillon, I. S.:Generalized nonnegative matrix approximations with Bregman divergences. In Ad-vances in neural information processing systems pp. 283-290 (2006).
7. Kingma, D. P., Welling, M.:Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013).
8. Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).
9. Mao, X., Shen, C., Yang, Y. B.:Image restoration using very deep convolutional encoder-decoder networkswith symmetric skip connections. In Advances in neural information processing systems pp. 2802-2810(2016).
9. Mao, X., Shen, C., Yang, Y. B.:Image restoration using very deep convolutional encoder-decoder networkswith symmetric skip connections. In Advances in neural information processing systems pp. 2802-2810(2016).
10. Huang, G. B., Ramesh, M., Berg, T., & Learned-Miller, E. (Labeled faces in the wild: A database forstudying face recognition in unconstrained environments (Vol. 1, No. 2, p. 3). Technical Report 07-49,University of Massachusetts, Amherst (2007).
11. Maaten, L. V. D., & Hinton, G. Visualizing data using t-SNE. Journal of machine learning research,9(Nov), 2579-2605(2008).
12. Baldi, P. Autoencoders, unsupervised learning, and deep architectures. In Proceedings of ICML Workshop on Unsupervised and Transfer Learning (pp. 37-49)(2012, June).
13. Krizhevsky, A., Sutskever, I., & Hinton, G. E. Imagenet classi_cation with deep convolutional neuralnetworks. In Advances in neural information processing systems (pp. 1097-1105)(2012).
14. LeCun, Y. (2015). LeNet-5, convolutional neural networks. URL: http://yann. lecun. com/exdb/lenet, 20.