Image Compression for Faster and Efficient Transmission Using RNN and Auto encoder Deep CNN Combined Model

Akash Ghosh
School of Data and Sciences
Brac University
Dhaka, Bangladesh
akash.ghosh@g.bracu.ac.bd

***Abstract*—Transferring image more faster and minimum usages of data is the main vision of this paper. Image compression becomes a very convenient path for solving the issue. In this paper,image compression with the vision of transferring images faster with minimum possible data loss. The goal of the project is to transfer image data more fast with smaller size even compromising the quality of the picture as less as possible .As many communication platforms are now offering free data transfer, but they are not offering free image transferring service. Through this initiative the problem of sending important images at the hectic time when someone is running out of data can easily find relief at the time of emergency not with the exact high quality, but they will get the image. During the research, this is kept in mind that the size will be reduced as much as possible and the sacrifice of the quality of the image will be done as little as possible. The expected outcome here is to reduce the size as little as possible so that it becomes possible to transfer the image within a shorter time and using less amount of data.**

***Index Terms*—data , image compression , quality , transfer images , package size , vision**

I. Introduction

This idea came to our mind in 2013 when we learned that there is a service on facebook which is free and can only be used for sending messages. In case of sending an image ,a person had to buy an internet package which was not convenient and if this is a hectic moment then it becomes more problematic . This issue used to bother us a lot and now it feels like the problem can be solved if the size of an image can reach nearly the higher limit of sending data for free. In this case, we have to sacrifice the quality of the image.

Here, in the project, our main target is going to be compressing the image as much as possible so that the data reaches the highest limit of sending data for free. The audience who will be benefited from this project can be from any background who is in emergency with time and need to send images on an emergency basis but facing trouble in connecting with the required speed of the internet for sending images. The project is hopefully going to take from 1.5 months to 2 months for full accomplishment.

We will be showing the comparison of the size of the image and comparing the quality of the image with the previous version. Any kind of random image will do for me with the project and the dataset images we are willing to get is from mnist datasets. The data will be initially jpeg format and then the data will be compressed initially using RNN model and then the compressed output will turn into numpy array and then fed into Auto encoder Deep CNN model for further compression. Google Colab is required for devoloping the model and then the models are to be installed to the end devices for performing the scenerio into reality.

We are here considering jpeg data type because there exists many formats to store images, among which JPEG is the most popular . JPEG is used by almost all image-capturing devices today. In 2015, 7 billion images were produced in JPEG format every day , which is much higher now. Total image format from 2022 to 2023 is expected to increase by 10.7%.Besides, 74.2% of the websites use JPEG as their image for.this is the mostly used data type [1] and because We are considering camera clicked image mostly, so, jpeg type data will be the best type and will make the maximum good in general. The final data analysis will be compared using graphs . As any kind of image should be able to be compressed, so hopefully data will be available easily. This is going to be a cloud based service.As this mainly focuses on the problem of transferring image within shorter time and using low amount of data, this can be implemented into communication machine which are nowadays cloud based.

II. Related Words

For the research purpose, first we needed to find which works have been done regarding this issue and found that some work using hardware and cloud [1] have been done so far which are pity much expensive and not so much time efficient. Those processes have risk of braking down the whole system for a little malfunctioning. So here any hardware and cloud device would not do the job in a safe and efficient way.

From studies [2] , this is found that only Context-Based Convolutional Entropy Modeling have been used for image compression and further image retrival. Our main focus in the paper is to minimizing the size of the bit stream as much we can.

In another study [3],authors have focused on model to make capable of reconstructing high-quality images even in low sampling ratios. In their proposed SCM for image CS, local connection priors are used during the sensing phase. It has been demonstrated empirically that SCM may maintain spatial characteristics, preventing block artefacts and high frequency noise in the final reconstruction. They have developed a novel LRKB to achieve higher reconstruction quality, by reformulating the process of image reconstruction as a discrete dynamical system. The LRKB can adopt highly efficient algorithms from ODE such as Runge-Kutta methods for numerical solutions. A RK-CCSNet implementation has attained cutting-edge performance in relation to prominent benchmarking baselines. A revolutionary system for processing sensor data has been created by introducing a novel end-to-end framework to encapsulate the modules. Here as they have used SCM, it allows linear operation. One advantage of the Scaled Composite Matrix Method is the preservation of an image's spatial characteristics (SCM). SCM sees the original image by gradually compressing the image size across a number of filters rather than using a single shared weight matrix multiplication for each pixel. The Scaled Composite Matrix Method (SCM) is a new way of analysing digital images. SCM sees the original picture by gradually compressing the image size over a number of filters, rather than using a single shared weight matrix multiplication for each pixel. They have used a dataset BSDS500 for 400 images for training and 100 data for testing. They have resized the images into (256,256). They have converted the image into converted to YCbCr color space and only the Y channel is used as the input of all the models.

In the coding section of the paper [3], authors have used Adam optimizer for training and set the exponential decay rates to 0.9 and 0.999 for the first and second moment estimate. The batch size is set to 4 and both CSNet+ [29] and RK-CCSNet were trained for 200 epochs at all, with the initial learning rate of 1e 3 and decay of 0.25 at 60, 90, 120, 150 and 180 epochs respectively. The sampling ratio for testing was set from 1/64 to 1/2, i.e., 1.5625 percent, 3.1250 percent, 6.2500 percent, 12.5000 percent, 25.0000 percent, and 50.0000 percent. PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural SIMilarity) [14] are chosen as the evaluation metrics throughout our experiments. Torch,math have been imported. Conv2d layers have been set according to the sensing rate.

Another study [4] combines entropy models with the learned DWT.which increases compression performance further. All entropy models, with the exception of one, including the learnt DWT, can easily be parallelized, and the systems offer reasonable encoding and decoding timings on GPUs.They have used 4 earned lifting based DWT architecture for their model. Learned lifting based (LLB) 2-D wavelet transform (WT) is constructed with the lifting scheme comprising neural network based predict and update filters. The DWT subbands are processed with a scaling neural network to scale the subband coefficients prior to scalar quantization.

But using DWT models for image compression is problematic. computational complexity and time,suitable mother wavelet, number of decomposition levels are some problems while working with DWT algorithm [5] [6]

we have combined the compression model of RNN and Auto Encoder Deep CNN because RNN is recurrent network, which feeds the result back to the network and suitable for image type data. on the other hand, CNN works as a Feed-Forward network and uses filters and pooling. This model is also suitable for image compression type job.

Another paper proposes a model Context-Based Convolutional Entropy Modeling for image compression. In the paper [7], The system they have used is Context-Based Convolutional Entropy Modeling. This simulation was needed because DNN based transforms are mostly non-invertible and encourage discarding perceptually unimportant image features during transformation. For context-based adaptive binary arithmetic coding (CABAC) , it considers the 2 nearest codes as partial context. So the size of PTX becomes large and becomes difficult to estimate conditional probability. PixelRNN and PixelCNN take advantage of DNN for modelling long range relations to increase the size of partial context, but are computationally intensive. So they are inefficient as well. In the proposed model, they have combined the CNNs with arithmetic coding for entropy modelling. The input grayscale image to eight binary planes and train a CCN and gets optimised in information theory. In case of loosy image, For lossy image compression, they have parameterized the categorical distribution with a discretized mixture of Gaussian (MoG) distributions, whose parameters are estimated by three CCN layers.The CCN-based entropy model is jointly optimized with analysis and synthesis transforms over training images, trading off the rate and the distortion. The entity they have used are Grayscale Image, Binarized Image Planes, input image , compressed image. Their attributes are Context-Based Convolutional Network layers , Arithmetic Encoder, Mean Estimates of Bernoulli Distributions. Their resources are Code Block , probability , partial context.Authors have tested their model on 2 datasets named Kodak and Tecnick datasets and measured the bpp(Bits per pixel) and efficiency of process time and they got the model is outperforming compared to the other model. If the number of parallel layers of the transformers could be increased, the model would work more efficiently and faster.

In comparison with previously worked models, the proposed model is out- performing in almost all the cases and the main goal proposed earlier to retrieve the maximum amount of data after image compression is also being served . For the lossless case, our method achieves the best compression performance. For the lossy case, the method offers improvements both visually and in terms of rate-distortion performance over image compression standards and recent mod- els which are being proposed.

For code [7], the input of the network (”data”) is a 3d map with 8 channels. The first mask conv layer ”conv1” takes the network input ”data” as input and adopt the ”contrain: 5” to select the mask for input layer as shown in Eqn. (10). And the second conv layer ”conv2” takes the 3 feature blocks generated by ”conv1” as input and adopt ”constrain: 6” to select mask for hidden layer. In the folder of ”cmp”, they provide the scripts to train and test the model for lossy image compression. And the pretrained models are available for testing. For training the model, you should train a base model with trainbase.py and a corresponding Gaussian mixture model(GMM) entropy model for the base model to make sure the GMM entropy model fulfil with the base model. Then, initialise the whole model with the base model and GMM entropy model, end-to-end train the model with teh ”traing mm.py”. To generate the bit stream, they adopt a post process step.

III. Research Methodology

The analysis transform generates the latent representation of the raw image while the synthesis transform maps the quantized latent back to the reconstructed image. To achieve a higher compression performance, the latent is modelled as Gaussian distribution in our entropy model and two hyper prior networks are used to estimate the entropy parameters mean and variance of the distribution, respectively. The two hyperpriors are used to estimate the entropy parameters and then the estimated entropy parameters are utilised to recover the quantized.The proposed method incorporates analysis and synthesis transform, adaptive quantization and entropy model. The quantization network Qz generates the quantization steps deceptively, which is then quantized to form the quantized latent . Encoder then uses estimated entropy parameters to compress and transmit the quantized latent representation. Firstly, the analysis transform E maps a block of one image x to the latent representation z. Finally, synthesis transforms maps the latent into the reconstructed image. It is worth noting that quantization steps,two quantized hyperprior are also transmitted as side information. On the decoder side, The quantization step s is first recovered to decode the hyperprior.

Then comes the part of CNN. An autoencoder is a neural network with an encoder parametrized by θ, that computes a representation Y from the data X, and a decoder gd, parametrized by ϕ, that gives a reconstruction ˆX of X. Autoencoders can be used for denoising or dimensionality reduction. If an autoencoder has fully-connected layers, the number of parameters depends on the image size. When it is used for compression, the representation is also quantized, leading to the new quantized representation ˆY = Q (Y). In order to create a rate-distortion optimization, the authors add the minimization of the entropy of the quantized representation. The encoder is followed by the normalizations at the encoder side, parametrized by ϕe. Similarly, the normalizations at the decoder side, parametrized by ϕd, followed by the decoder. In this case, Y ϵ Rh×w×m is a set of m feature maps of size n = h The basic autoencoder training minimises the image reconstruction error. This implies that one autoencoder has to be trained per image size. To avoid this, an architecture without fully-connected layers is chosen. It exclusively comprises convolutional layers and non-linear operators. Moreover, a bit allocation is performed by learning a normalisation for each feature map of Y.

A. RNN Compression Model

Recurrent neural networks (RNNs) are a form of neural network in which the results of one step are fed into the current stage as input. Classical neural networks have inputs and outputs that are independent of one another, but there is a requirement to remember the previous words in situations when it is necessary to anticipate the next word in a phrase. As a result, RNN was developed, which utilized a Hidden Layer to resolve this problem. The Hidden state, which retains some information about a sequence, is the primary and perhaps most significant characteristic of RNNs.

Two convolutional kernels are employed in the recurrent units that make up the encoder and decoder: one on the state vector, which gives the unit’s recurrent nature, and the other on the input vector, which enters the unit from the previous layer. The state vector’s convolution and its kernel will be referred to as the ”hidden convolution” and the ”hidden kernel,” respectively. Along with the output depth, we also include the input-vector convolutional kernel’s spatial range. Full mixing across depth is possible with all convolutional kernels. Except for units D-RNN#3 and D-RNN#4, where the hidden kernels are 33, the spatial extents of the hidden kernels are all 11. In comparison to 11 hidden kernels, the bigger hidden kernels consistently produced better compression curves.

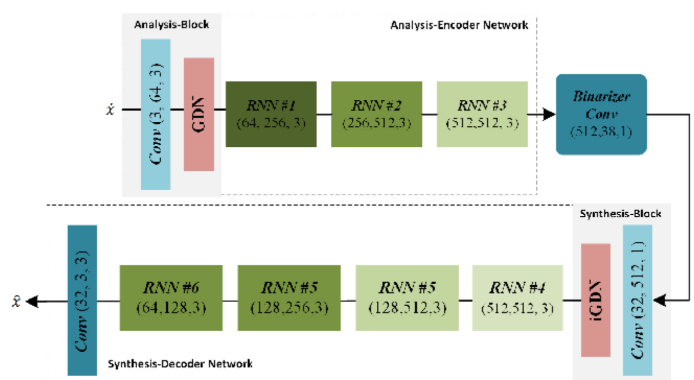


Fig. 1: Iterative Architecture of RNN

B. CNN Compression Model

A part of machine learning is convolutional neural networks (CNNs). It is one of the many distinct kinds of artificial neural networks that are used to numerous applications and data sources. Specifically used for image identification and other tasks involving the processing of pixel data, CNNs are a type of network design for deep learning algorithms.

Here Deep CNN Autoencoder model have been used for the further compression of the image.Dense layer,Convulation layers,Maxpooling layers are available in this part of the compression model.At the very first, after converting the image, the have been divided by 255 for normalizing the value of the pixel.then the input data have been reshaped to the dimension of 28*28. Now the stream will go through the model.

In the Model,the bit stream will go through the the convulation2D+ReLU layer and then the Max pooling layer. After encoding the data, the stream will go through the decoding block.

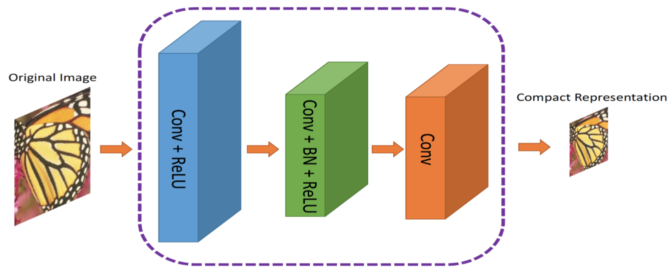


Fig. 2: Architecture of CNN

C. Our proposed Combined model

In our model, the image first goes through the RNN model and we get an image as an output and after iteration, the image converts into numpy array to go through the CNN model and get compressed again. We hope the model performance to work in the following sequence as the figure.

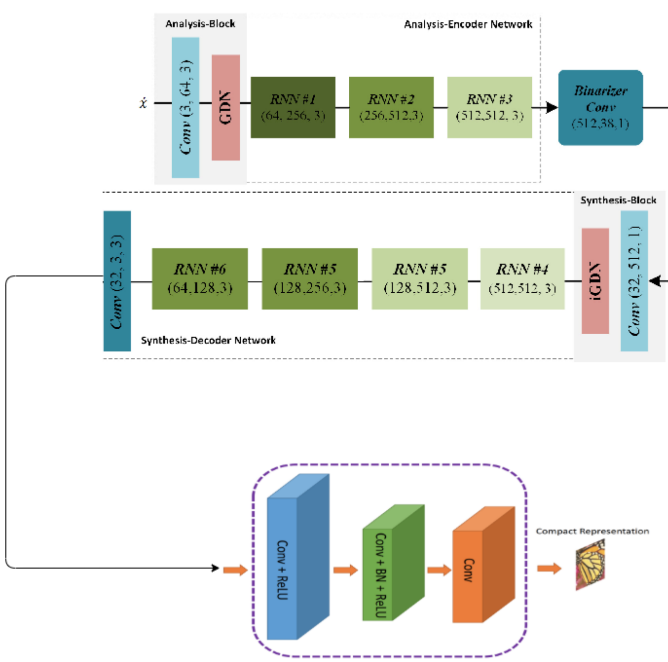
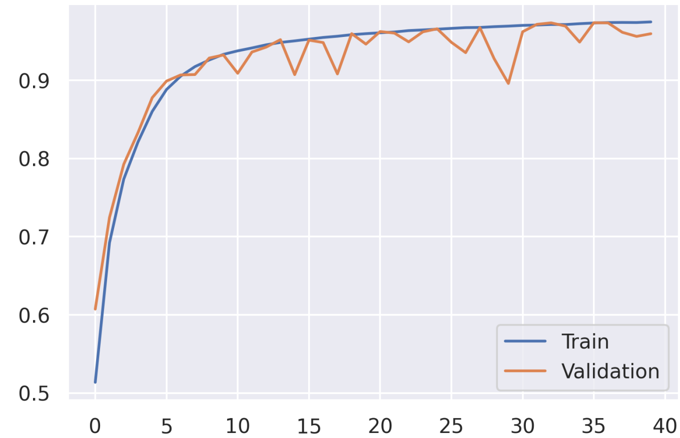


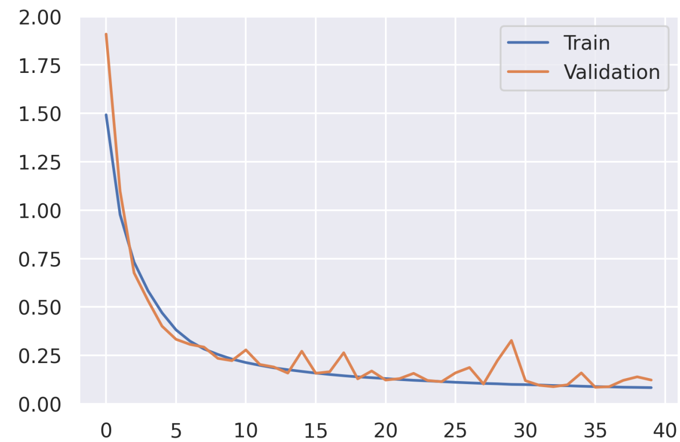
Fig. 3: Proposed model skeleton

IV. RESULTS

We have used mnist data set for the whole process. after training and testing of the data,the RNN algorithm shows Accuracy per epoch and loss per epoch as follows :



(a) Accuracy Per Epoch(RNN)



(b) Loss Per Epoch(RNN)

Fig. 4: Epoch Values (RNN)

Then we evaluated the CNN Model and found the following epoch result.

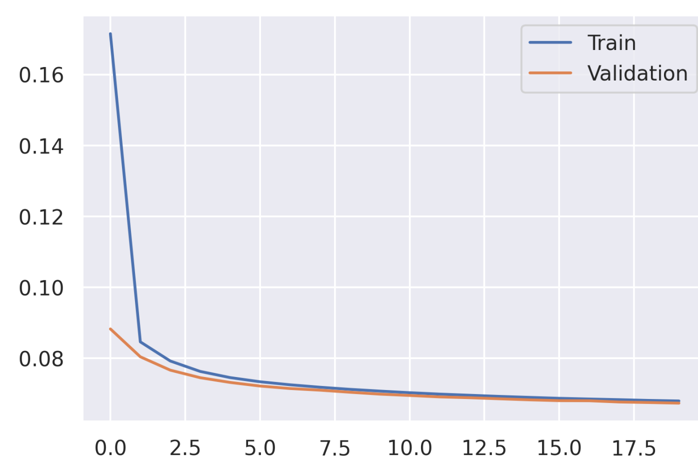


Fig. 5: Loss Per Epoch(CNN)

After evaluating the result, this can be found that the original image was of 290 KB in size and the compressed image is of size 61.8 KB which means a size compression of 78.69% have been possible with this model. The quality of the image have been preserved at maximum possible level by using the balanced number convulation and relu layer. The image comparison can be seen from the figure. The Similarity using SSIM is: 0.7125490511339234 which means 71.26% pixel of the image matches with the originaly compared one.

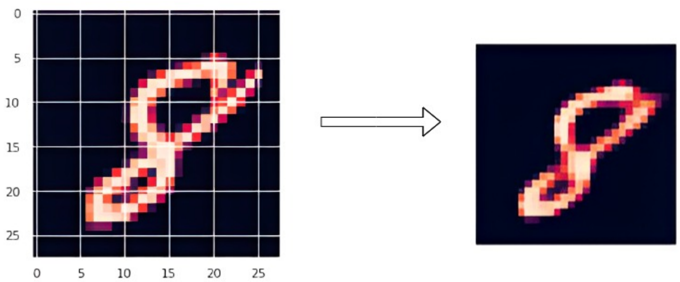


Fig. 6: input vs output

For further clarification, we have run the model again and got similar compression rate.

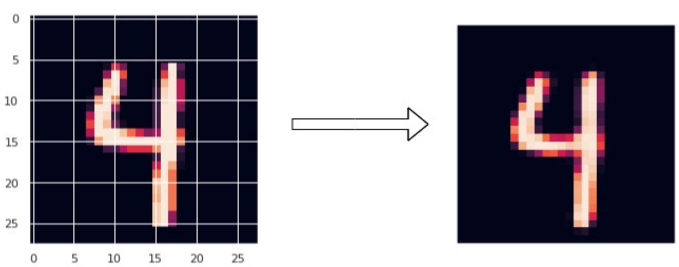


Fig. 7: input vs output

V. Conclusion

In the paper the main focus of investigation was to make the image size as much as possible. This would make the whole communication process easier and smoother. If image could be compressed with lowest amount of loss,the image could still be understandable. May be the image looses it's quality and can not be used for sophisticated uses but it is still understandable and work can be done. So in this paper we have focused to protect the quality of the picture as much as possible and then compressed it. As from the above discussion, this can be found that this is an working model for image compression. We have plan to try to model for Video compression as well and try to implement into search engine and communication app through which , communication can become more faster, cheap and spread.

References

[1] J. Noor, M. N. H. Shanto, J. J. Mondal, M. G. Hossain, S. Chellappan, and A. B. M. A. A. Islam, "Orchestrating image retrieval and storage over a cloud system," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2022.

[2] M. Li, K. Ma, J. You, D. Zhang, and W. Zuo, "Efficient and effective context-based convolutional entropy modeling for image compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 5900–5911, 2020.

[3] R. Zheng, Y. Zhang, D. Huang, and Q. Chen, "Sequential convolution and runge-kutta residual architecture for image compressed sensing," in *European Conference on Computer Vision*. Springer, 2020, pp. 232–248.

[4] U. Berk Sahin and F. Kamisli, "Image compression with learned lifting-based dwt and learned tree-based entropy models," *arXiv e-prints*, pp. arXiv–2212, 2022.

[5] U. Singh, "What are the demerits of using the discrete wavelet transform (dwt) and the singular value decomposition (svd) for extracting discrete time feature?" 10 2018.

[6] M. M. Feraidooni, "What are the demerits of using the discrete wavelet transform (dwt) and the singular value decomposition (svd) for extracting discrete time feature?" 10 2018.

[7] M. Li, K. Ma, J. You, D. Zhang, and W. Zuo, "Efficient and effective context-based convolutional entropy modeling for image compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 5900–5911, 2020.