LOSSLESS IMAGE COMPRESSION FOR MEDICAL IMAGES USING NEURAL NETWORKS

Bachotti Sai Krishna Shanmukh
Department of Electrical Engineering
Indian Institute of Technology Madras
Chennai, India - 600036
ee19b009@smail.iitm.ac.in

July 10, 2022

ABSTRACT

Medical Images are used in healthcare to diagnose certain health conditions and often require a very accurate pixel map for computer-aided diagnosis. These images are also used as datasets for both supervised (or) unsupervised deep learning models to diagnose critical health conditions, analyze the nature and spread of tumors etc. Hence, compressing these images in a lossless way i.e., capturing the complete information present in the original pixel map into a lower file size configuration is critical for accurate diagnosis. In this paper, we discuss about using Neural Networks for lossless image compression.

Keywords Lossy Compression · Lossless Compression · Convolutional Neural Networks · Auto-encoders · Variational Autoencoders (VAE) · Conditional Generative Adversial Networks (CGAN) · Delta · Multi-Scale Structural Similarity Index (MS-SSIM) · Bit-rate

1 Introduction

Image compression is a widely used idea to store information present in an image, in a lower file size than original. They are essentially are of two types, Lossy and Lossless. Some of the well known general purpose lossless image compression algorithms include Arithmetic encoding and Huffman encoding. These algorithms are based on entropy coding. Later, a compression technique called Discrete Time Cosine transform was used for image compression. In 1997, two of the popular compression techniques, JPEG2000 and PNG were developed. JPEG 2000 is a compression standard based on a discrete wavelet transform. It has both lossy and near lossless formats, while the later used reversible integer wavelet transform. PNG on the other hand is a completely lossless image compression algorithm

Neural Networks for Image compression can offer higher compression rates but give lossy reconstructions. Instead of using cosine transformations or entropy encoding, Neural Networks primarily work on the principle of combination of Encoder neural network (or a compressor) and a Decoder neural network (a generator). The compressor learns to represent the input image into a latent code of a lower dimensional space, also called as latent space. The generator learns to reconstruct the image back from latent space to the input space. There is a distortion between the input image and the reconstructed image.

2 Literature Review

The lossy JPEG 2000 compression technique neglects the finer details perceivable by human eye to achieve a lower bi-trate. An optimized JPEG 2000 compression approach for Whole Slide Images (WSI) has shown upto 20:1 compression without loss of any pathological information[1].

Residual GRU [2], uses an architecture called Residual Gated Recurrent Unit, which works iteratively on reconstruction image until it is refined to a high perceptual quality image. Although the compression rate look attractive, it is a lossy

Figure 1: Above images are zoomed in versions. At original resolution, Left: Original image (1419 KB PNG) at 1.0 MS-SSIM. Center: JPEG (33 KB) at 0.9 MS-SSIM. Right: Residual GRU (24 KB) at 0.9 MS-SSIM. This is 25% smaller for a comparable image quality

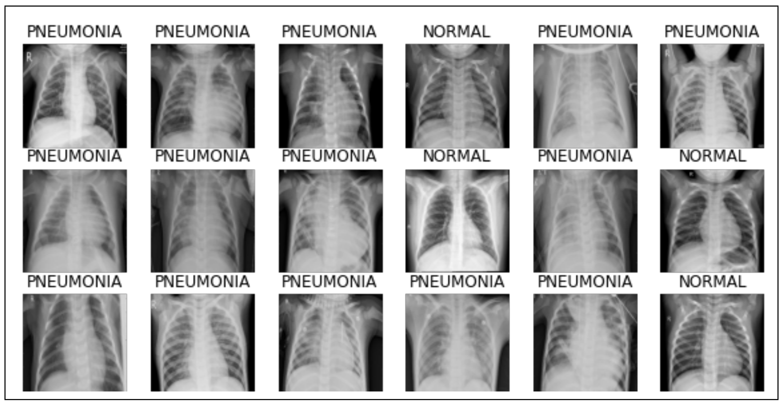

Figure 2: Sample Chest X-rays from Dataset

compression technique and cannot be used for medical images. We can observe that the finer details like whiskers and texture (as shown in Fig.1) are lost for the Residual GRU reconstruction image

Initial works in Image compression relied on RNNs, while subsequent works were based on auto-encoders. In [3] Ballé $et\ al.$, demonstrate Variational autoencoder based image compression with a scale hyperprior, where a prior distribution over the latent space is assumed and the conditional distribution of output space over the latent space is learnt. The latent space is regularized and well-learnt, however the reconstructed images are blurry, as all pixels are assumed to be independent random variables for a given latent code. Also, the generator works on sampling from the learnt probability distribution. Hence, it is not suitable for lossless compression as the generator is not a deterministic network.

Recent work by Blau and Michaeli [4] showed the existence of a triple "rate-distortion-perception" trade-off, formalizing "distortion" as a similarity metric comparing pairs of images (like d(x,x') where d is MSE), and "perceptual quality" as the distance between the image distribution p_X and the distribution of the reconstructions $p_{X'}$ produced by the decoder, measured as a distribution divergence. They show that at a fixed rate, better perceptual quality always implies worse distortion. Conversely, only minimizing distortion will yield poor perceptual quality.

In [5], Agustsson $et\ al.$ demonstrated the potential of using GANs to prevent compression artifacts with a compression model that produces perceptually convincing reconstructions for extremely low bitrates (<0.08 bpp). However, their reconstructions tend to only preserve high-level semantics, deviating significantly from the input. Also, GANs need dataset of size atleast 50,000 images and contains a potential chance of mode collapse, where the generator produces only certain subclass of inputs. In [6],Fabian $et\ al.$ demonstrate $HiFiC$, a lossy compression system using CGANs. They bridge the gap between rate-distortion-perception trade-off, by evaluating both quantitatively with various perceptual metrics, and with a user study.

3 Method

In medical images, the anatomy is similar across different records.In this paper we use Neural networks to capture primary image information at a low bit rate. The difference between image and reconstruction (termed as delta) is then

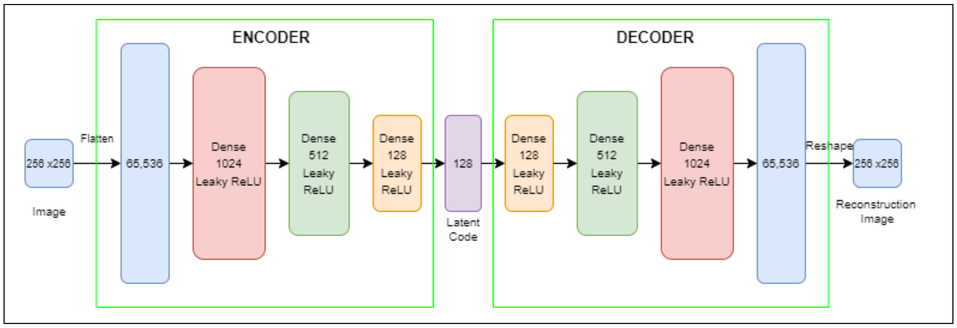

Figure 3: Architecture of ANN based Auto-encoder used.

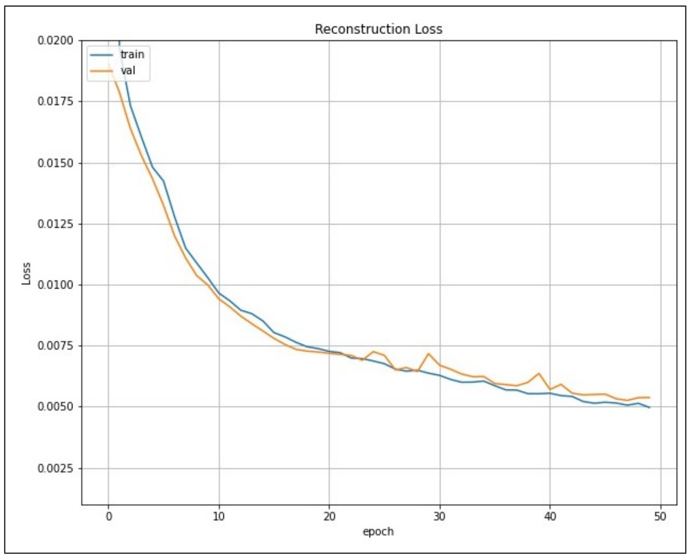

Figure 4: Training curve of the ANN based Auto-encoder with MSE as the loss function. Best weights are saved at the end of 48 epochs. On the validation set, MSE = 0.00525

stored in a lossless way by using Arithmetic encoding, Huffman encoding or DCT. In this paper we discuss variants of auto-encoders trained on a custom dataset and also compare the bit-rate of lossless algorithms for storing the delta. At the end we also compare the results of auto-encoders with $HiFiC$ [6] model.

3.1 Dataset and Training

To demonstrate proof of concept, a dataset of chest X-ray images [8], containing 5,866 images is used. The images contain healthy and pneumonia affected chest X-rays. About 90 $\%$ is used for training and the remaining for validation. Since, it is an unsupervised learning problem, we do not need the class labels. For uniformity, we resize the images to 256×256, single channel (gray-scale) 8-bit color format. The code for models is written in Tensorflow Keras and Google Colab free GPU version is used for training the models

3.2 ANN based Autoencoder

A neural network containing only dense layers is used as an initial approach. The dense layers are followed by Leaky ReLU activation unit. The encoder maps from an input space of (65536) to a latent space of (128). A total of 135,466,112 trainable parameters are present in the entire encoder-decoder network. We chose MSE (Mean Squared Error), a metric that measures the distortion between input image and the reconstructed image, as the loss function. The training was done with a batch size of 32 images and stopped after 48 epochs.

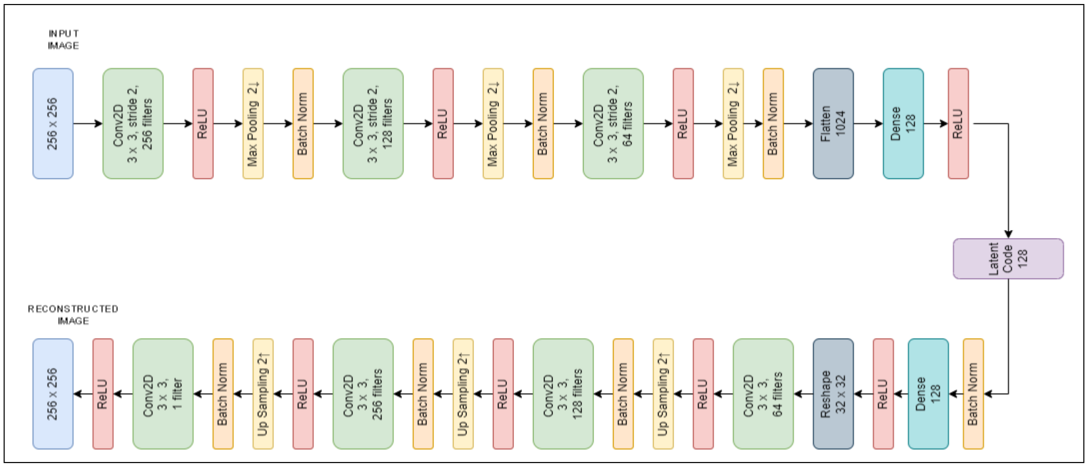

Figure 5: Architecture of CNN based Auto-encoder used

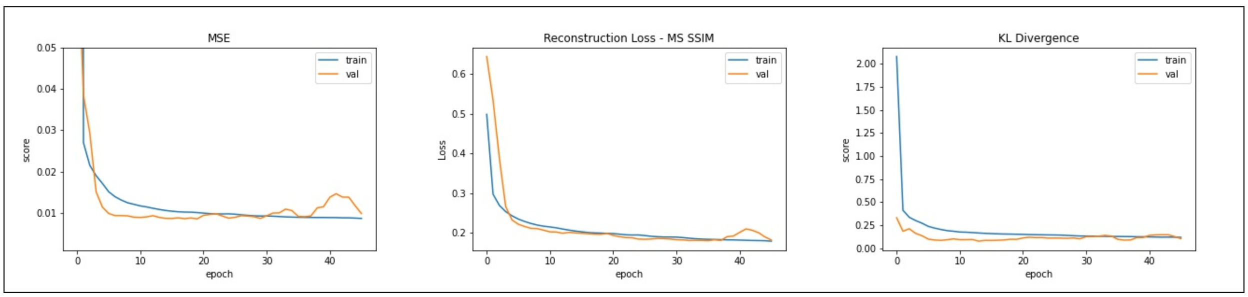

Figure 6: Training curves for CNN approach 1.0. The best weights are saved at the end of 36 epochs. On validation set, MSE = 0.01003, MS-SSIM Index = 0.8027, KL Divergence = 0.1385

3.3 CNN based Autoencoder

Convolutional Neural Networks learn the edges and patterns in an image more affectively with lower number of trainable parameters. Hence, the next approach was to try CNN based Autoencoder. The encoder model uses Max Pooling after every convolutional layer to map down to latent space. Whereas, the decoder model uses Up sampling to map back to image space. Batch Normalization layers are used between each nonlinear and linear operation for faster convergence. A total of 1,008,449 trainable parameters are present in the encoder as well as decoder. The architecture used is demonstrated in Figure 4.

3.3.1 CNN 1.0

To tackle the problem of blurry reconstructions, a different loss function is used here. MS-SSIM (Multiscale Structural Similarity Index) which indicates how close the reconstructed image looks compared to original image. An MS-SSIM score of 1.0 indicates that the reconstruction looks exactly same as the original image. We use a loss function defined as (1 - MS-SSIM) to train the CNN model. The total number of trainable parameters present in the model are 1,008,449. The model is trained for 36 epochs on a batch size of 32.

3.3.2 CNN 2.0

In an attempt to capture more information using Neural network compression, we now use a latent space of dimension 512 and train the model on a custom loss function. A total of 1,795,265 trainable parameters are present in the entire model. We define a custom loss function to train the model with an aim to lower the distortion and improve the perceptual quality.

$$L(x, x') = \lambda_1 \times MSE(x, x') + \lambda_2 \times (1 - MultiscaleSSIM(x, x')) \tag{1}$$

We chose the hyper-parameters $\lambda_1 = \lambda_2 = 0.5$ and trained the model for 26 epochs on a batch size of 32.

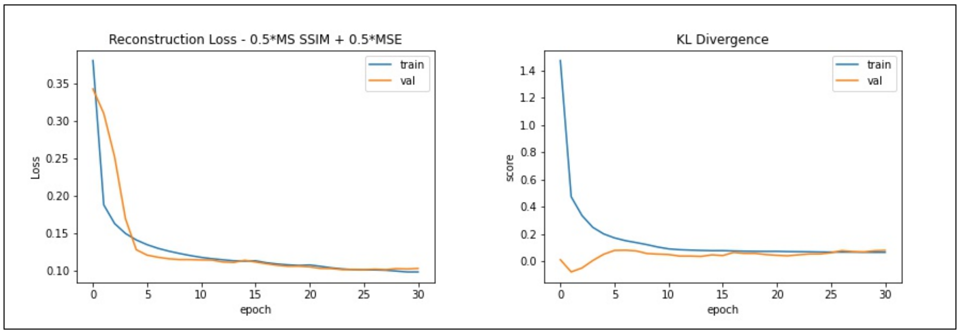

Figure 7: Training curves for CNN 2.0. The best weights are saved at the end of 36 epochs. On validation set, MSE = 0.00839, MS-SSIM Index = 0.7826, KL Divergence = 0.1082

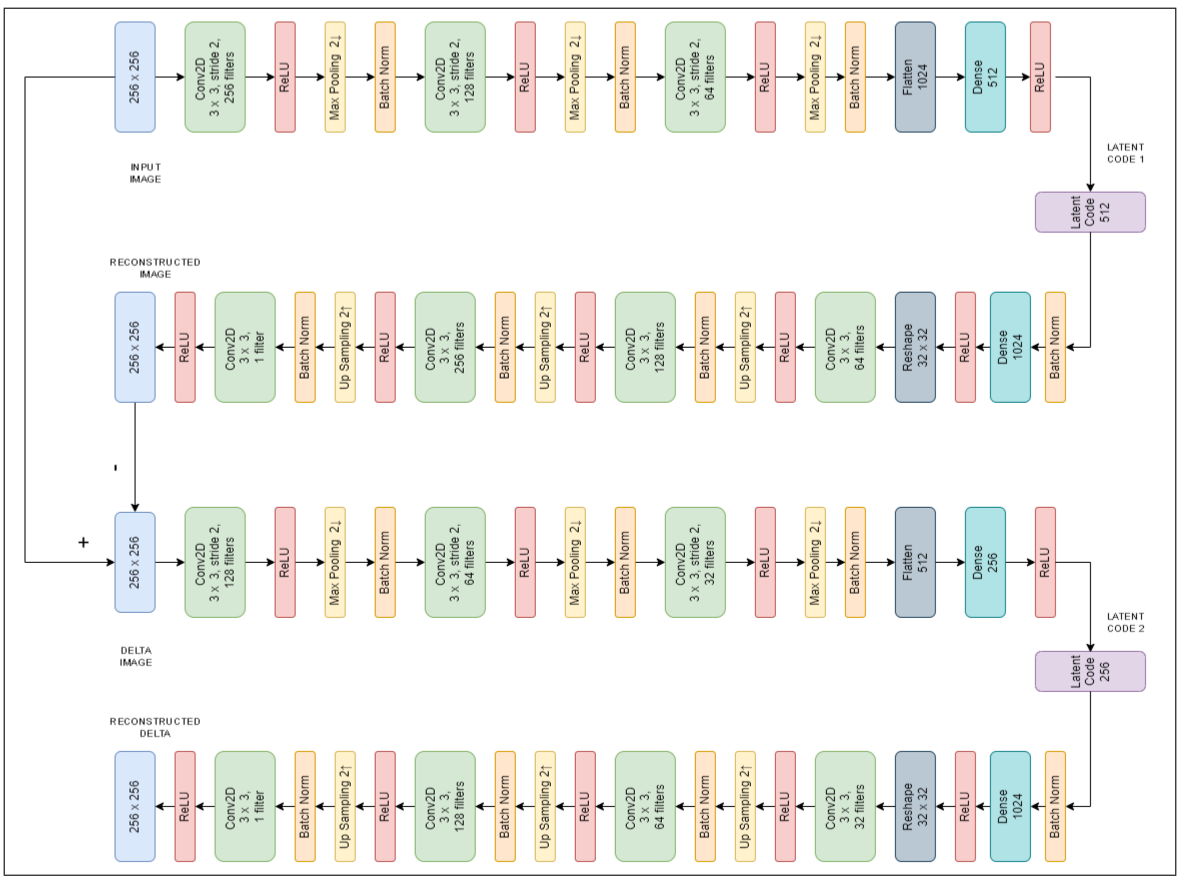

Figure 8: Architecture of Two Stage CNN Auto-encoder used

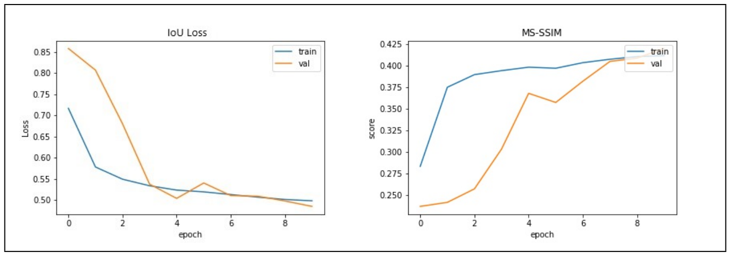

Figure 9: Training curves for second stage of Auto-encoder. The best weights are saved at the end of 10 epochs. On validation set, IoU is 0.5013, MS-SSIM is 0.5885, MSE is 0.0074 and KL Divergence is 0.0558

3.4 CNN 3.0: Two Stage Auto-encoder

We use the trained CNN 2.0 model as the first stage auto-encoder and build another auto-encoder to compress the delta (as shown in Fig.7). Since, mse and ms-ssim are already used as loss functions and the model training has saturated, we define a custom Intersection over Union metric and use it in the loss function to train the second stage.

$$CustomIoU(x, x') = \frac{x \cdot x'}{x \cdot x + x' \cdot x' - x \cdot x'} \tag{2}$$

$$L(x, x') = 1 - CustomIoU(x, x') \tag{3}$$

where x and x' are the original and the reconstructed images respectively. The second stage was trained for 10 epochs for a batch size of 32 with 583,265 trainable parameters.

3.5 Lossless Compression techniques

Since the end goal of the compression system is to compress the images in a lossless way, we compress the error (delta) between reconstructed image and the actual image. Since our decoder model computes in a deterministic way, the latent code and the compressed error together can reconstruct to original image. Huffman Encoding, Lossless Discrete Cosine Transform and PNG are the suitable algorithms that can be used. In Huffman encoding each pixel value is assigned a code based on the frequency (probability) of occurrence in the image. Higher the occurrence of a value, lower code size will be assigned. In this way a unique code is mapped to each value in a tree traversal manner.

4 Inference

4.1 Models

In the ANN based auto-encoder the validation loss (MSE) is 0.00526 for normalized images. We observe that the reconstruction images are blurry (in Fig. 11). This is caused due to choosing mean squared error as loss function, as it only optimizes on reducing the distortion between input and the reconstructed image. Here the perceptual quality of image is traded-off for lower distortion.

In the CNN 1.0 model, the validation loss is 0.1973 (i.e., a similarity of 80.27%, since MS-SSIM is used in the loss function). We observe that relying on MS-SSIM as loss function causes the model to give perceptually more pleasing images but poor texture reconstructions. The borders of main chest-frame are clearer in Fig.12 than in Fig. 11, but finer details like rib branches are not clearly reconstructed. Also the the distortion metric MSE is observed to be 0.01003, which is more than the ANN model. We also evaluate the model for KL-Divergence, a statistical value that measures how probability distribution of pixel values of original image is different from the second distribution and it has a score of 0.1385 on test set. Clearly CNNs play a better role in capturing more information in latent code despite using a perceptual metric as loss function.

The CNN 2.0 model shows an MSE value 0.0084 which is lower than the CNN1.0 model. Also the KL Divergence is lower than in the previous approach. In Figure 13, row 2 image, we can observe that the reconstruction image looks less cloudy at the chest-frame than in the Figure-11 counterpart. Also, the delta images have a higher background contrast for CNN 2.0 model (in Fig. 13), indicating that the reconstruction has captured out more information from the original image, than in case of CNN 1.0 model.

An IoU score 0.9686 is observed for the CNN 2.0 model on the test dataset (of size 580 images), whereas the previous CNN model has an IoU score of 0.9640 on the test dataset. The IoU score for ANN model is found to be the best among the three, i.e., 0.9813 indicating that the reconstructed image pixel values are less distorted from the actual image.

For the two stage model, the second stage has an MSE value of 0.0074 and custom IoU score of 0.5013. Not much information is extracted from the delta by stage 2. The input image of Stage 2 and the error image of stage 2 (in Fig.14) look almost identical with some distortion.

4.2 Lossless Compression and Bit-rate

The original image is an 8-bit color, single channel (gray-scale) image of size 256 $\times$ 256. Hence the bits per pixel used will be 8 bpp. The original image in .PNG format has a bitrate of 4.641 bpp. The error between original image and the reconstruction is also of size 256 $\times$ 256. Before we compress the image, we do a quantization step of the normalized error values. If we use 8 bits to store one pixel value in delta, we account for a precision of $\frac{1}{256}$. When the quantized value is used for reconstructing original image, we expect an integer round-off error of ± 1 (on absolute pixel value scale of 0 to 255). We can add more bits to store the delta to decrease the probability of occurrence of such errors, which is completely dependent on pixel error value. To avoid any type of loss, an alternative method is suggested. Instead of quantizing the error, we scale and quantize the output of the Neural Network model to an integer value between 0 and 255, compute delta and then compress.

4.2.1 ANN Model

In the ANN model, the reconstruction image is effectively represented by the float32 latent code of 128 size, which needs 4096 bits. But the information present in the delta is less than the original image, hence can achieve better compression. Applying Huffman encoding to quantized delta, the number of bits to store the delta reduces from 524288 to 448946. The bit-rate is 6.913 bpp (bits per pixel), which is higher than the original image bit-rate in .PNG format. When Huffman encoding is applied to error computed via the alternative method, the number of bits needed to store the delta is 407765 only i.e., a bit-rate of 6.284 bpp (lower than the previous approach) When the delta image is stored in PNG format, it occupies 323624 bits and the overall bit-rate is 5.000 bpp, which is higher than original image by 7.73%.

4.2.2 CNN 1.0 Model

The latent space is of dimension 128. Hence, 4096 bits are needed. The error between original image and the reconstructed image from CNN model is compressed. Huffman encoding on quantized delta reduces the number of bits to store to 380979. The bit-rate for the image is 5.875 bpp. When Huffman encoding is applied to error computed via the alternative method, the number of bits needed to store the delta is 428789 only i.e., a bit-rate of 6.605 bpp. Delta image stored in PNG format occupies 288640 bits, which is less compared to delta image of ANN model. The bit rate reduced to 4.467 bpp, which is lower than the bit-rate of original image by 3.75%

4.2.3 CNN 2.0 Model

Here the latent space is of dimension 512. Hence, 16384 bits are needed to store the latent code. Applying Huffman encoding on quantized delta images reduces the number of bits to store to 385834 i.e., a bit-rate of 6.137 bpp. Delta image stored in .PNG format occupies 286832 bits, which equals a bit-rate of 4.627, which is lower than bit-rate of original image by 0.3%

4.2.4 Two Stage Auto-encoder

Here, we have two latent codes of 512 and 256 size respectively, which take 24,576 bits. Applying Huffman encoding on the quantized delta of second stage auto-encoder reduces the number of bits to 376083. The bit-rate is found to be 6.113 bpp.

Delta image stored in .PNG format occupies 291112 bits, which is more than single stage models. The bit-rate to store image is 4.817 bpp which is 3.8% more than the bit-rate of original image. The second stage model is model unable to compress the delta of first stage, rather inducing distortion causing an increase in bit-rate of delta to be stored.

4.2.5 HiFiC : Pre-trained GAN

This model compresses the images and retains high fidelity as well. It compresses the image to a low bit-rate of 0.1438, but storing the delta takes 314440 bits and the overall bit-rate is 4.941. This can be attributed to the high fidelity added in reconstruction, causing distortion from the original image. Although the reconstruction looks very similar to original image, we can observe there is high error between images. (In Figure 10)

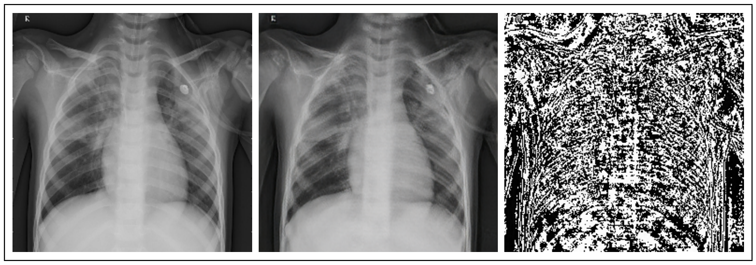

Figure 10: On Left: Original Image, In Middle: Reconstructed Image from HiFiC compression, On Right: Delta Error between images. Sharper noise is observed in delta indicating higher distortion

Table 1: Bit-rates observed on Original Image shown in Fig. 10

Model	Delta (bpp)	Overall Bitrate (bpp)
Original(.png)	-	4.641
ANN + .png	4.938	5.000
CNN 1.0 + .png	4.404	**4.467**
CNN 2.0 + .png	**4.376**	4.627
Two Stage AE + .png	4.442	4.817
$HiFiC$ Pre trained GAN + .png	4.797	4.941

5 Conclusion

The two CNN auto-encoder models have performed well in compressing the images compared to other models. The significance in the decoder models is they are deterministic and hence produce always the same reconstruction for a given latent code. CNN 1.0 has shown the least bit-rate overall compression. Although CNN 2.0 had lower bit-rate for Delta, the higher latent space dimension (512) has increased the overall bit-rate. In CNN 2.0, using the custom loss function which includes both MS-SSIM and MSE together showed reconstructions with better information and less noise compared to CNN 1.0. The second stage in two-stage auto-encoder could not capture the first stage delta, rather distorted it more, causing a higher a delta bit-rate than single stage auto-encoder models. The HiFiC compression (GAN), generates perceptually good images but has a higher delta bit-rate than the auto-encoder models. More exploration needs to be done on good loss functions and metrics to get better reconstructions. The quantization step can be added in loss functions for the model to learn pixel to pixel same integer values instead of integer round-off errors. A way forward would be to train an Encoder and a GAN together, where the GAN takes the latent code and class of the disease as inputs and generates an even closer representation of the original image. Literature shows that training with a GAN yields reconstructions that outperform other models at practical bitrates, for lossy high-resolution images. The challenges would be to choose a right architecture and loss function that balances out on reconstructing to a near-lossless image.

Link to code

References

1. Optimized JPEG 2000 Compression for Efficient Storage of Histopathological Whole-Slide Images
2. Full Resolution Image Compression with Recurrent Neural Networks
3. Variational image compression with a scale hyperprior
4. Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff
5. Generative Adversarial Networks for Extreme Learned Image Compression
6. High-Fidelity Generative Image Compression
7. Lossy Medical Image Compression using Residual Learning-based Dual Autoencoder Model
8. Chest X-ray Dataset

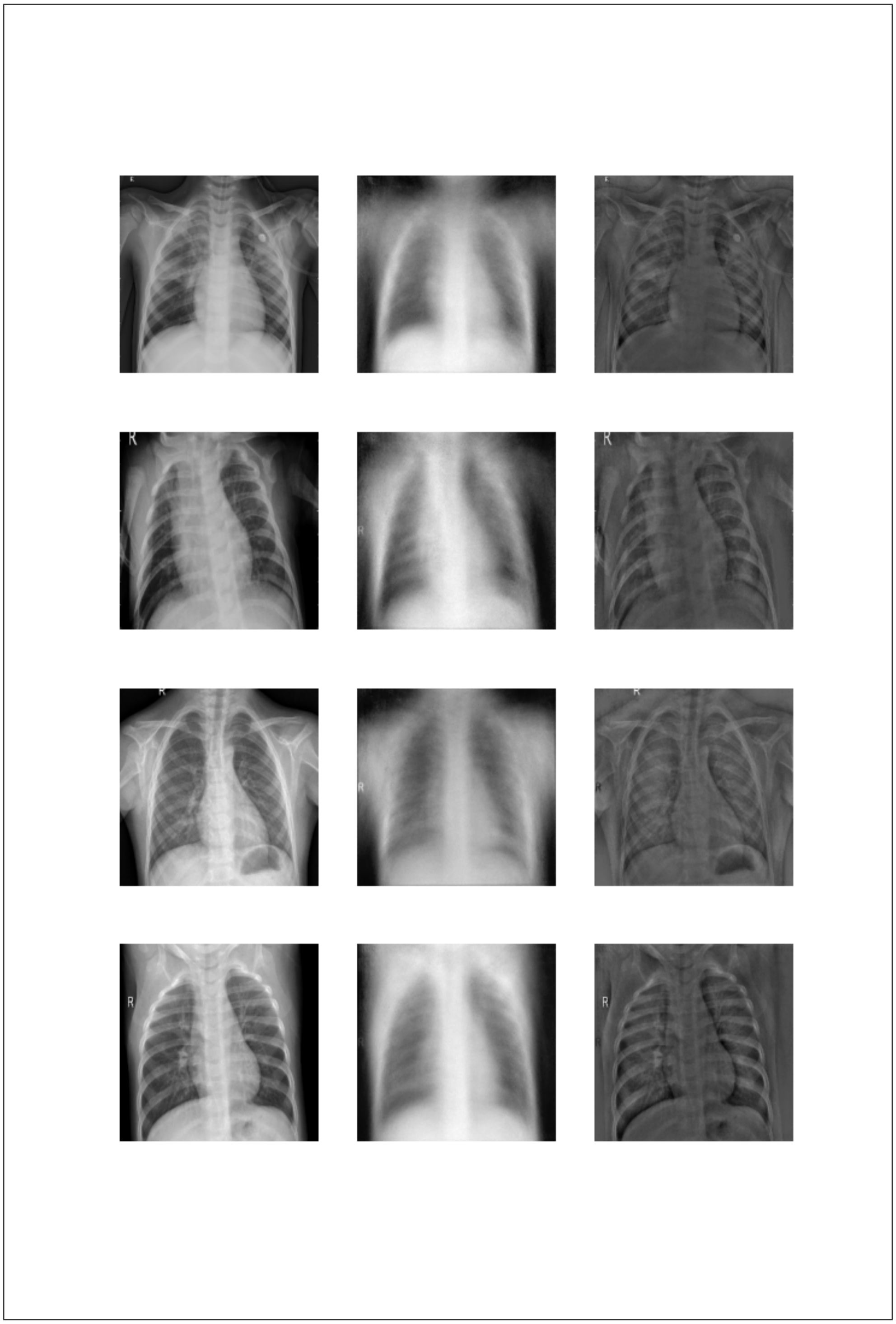

Figure 11: ANN model trained on MSE loss function. On the Left are the original images. In the Middle are the reconstructed images and on the right are the delta images

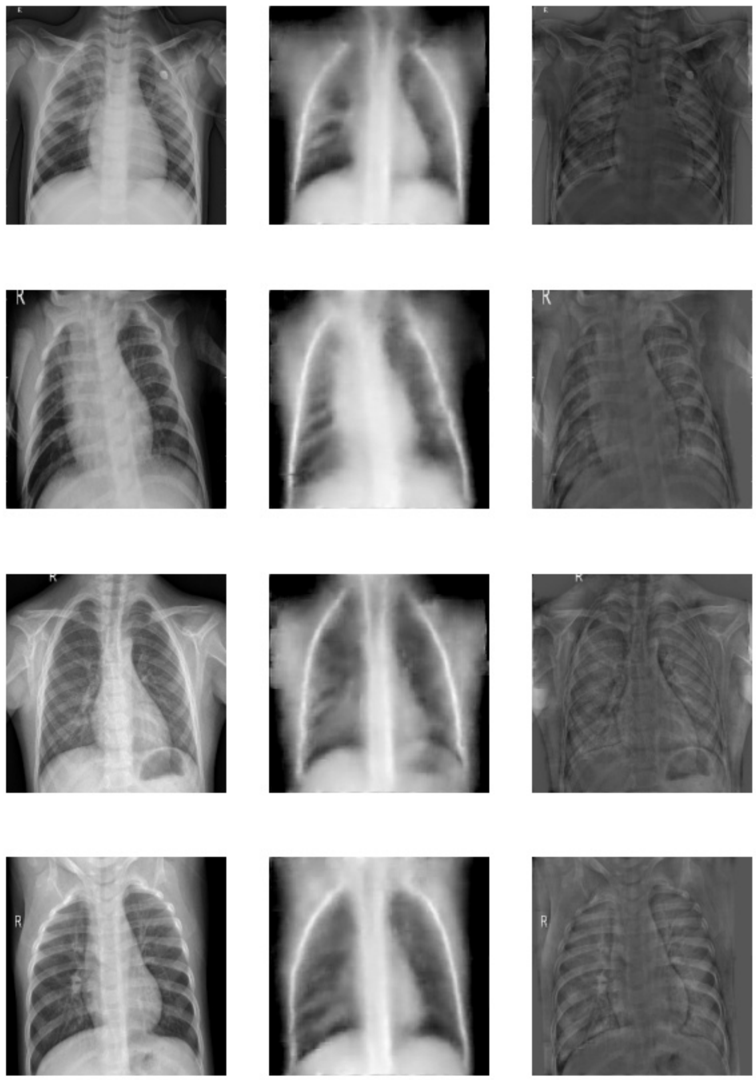

Figure 12: CNN model trained on MS-SSIM loss function. On the Left are the original images.In the Middle are the reconstructed images and on the right are the delta images

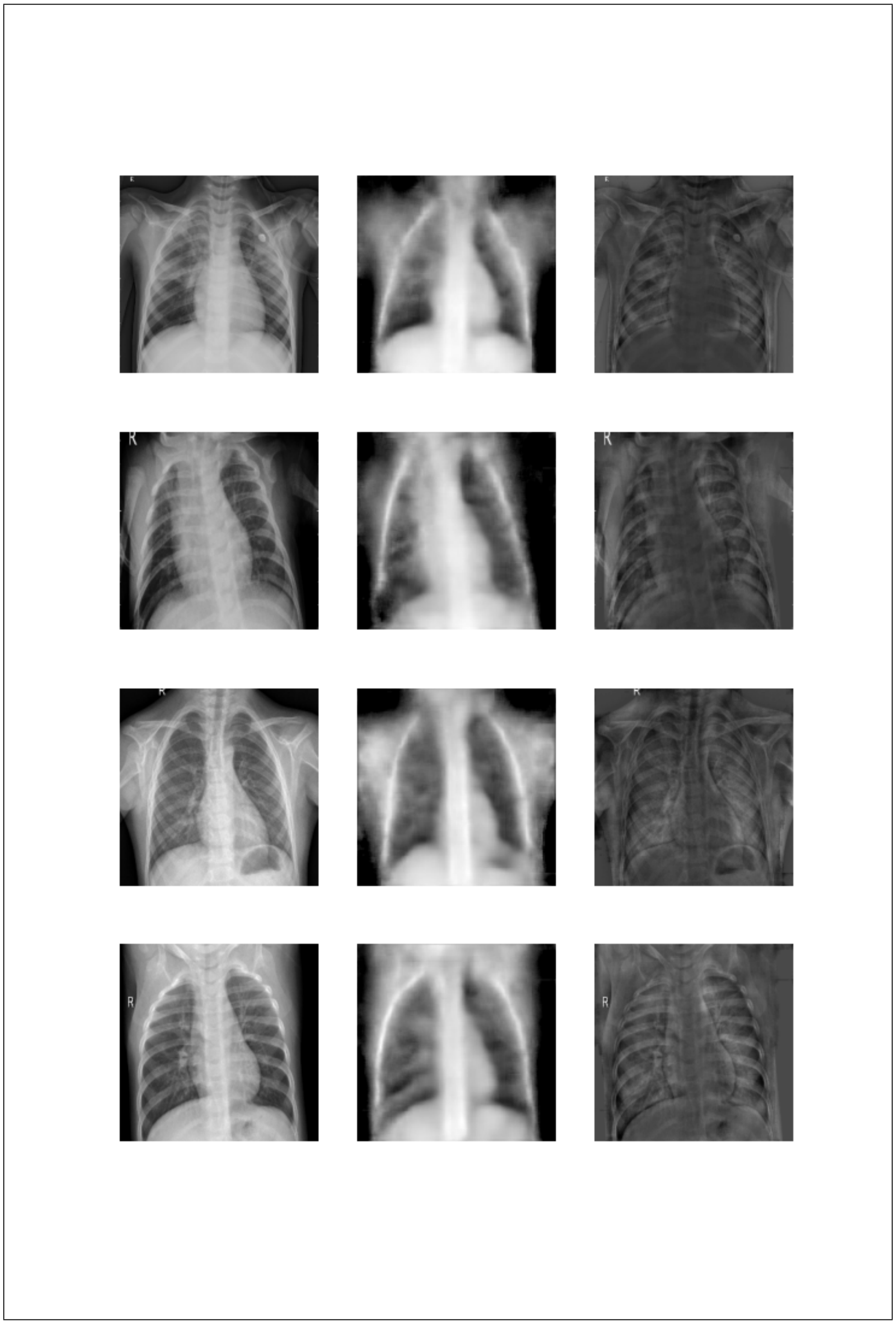

Figure 13: CNN model trained on custom loss function of MS-SSIM and MSE. On the Left are the original images. In the Middle are the reconstructed images and on the right are the delta images

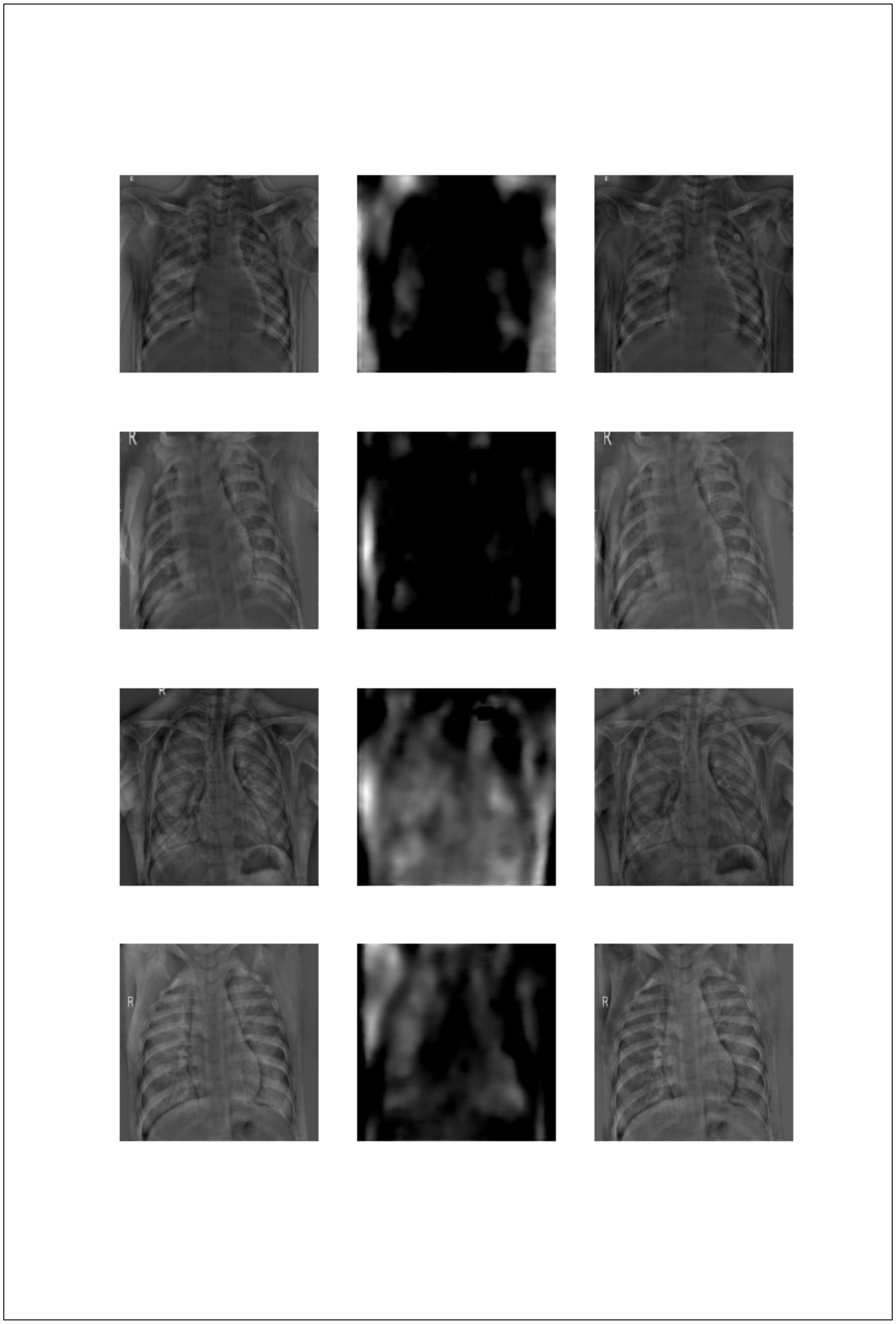

Figure 14: Two stage CNN based Auto-encoder, second stage trained on custom IoU loss function. On the left are the delta images of stage 1. In the middle are the reconstructions of the delta images and on the right are the difference of delta and reconstruction of delta