Data Compression via Nonlinear Transforms using Artificial Neural Networks

Group T4: Spencer Hill, Wyllie Schenkman, Jordan Curnew, Mark Benhamu

Supervisor: Dr. Tamas Linder

Final Presentation

Problem Definition and Application

Problem Definition

Compress digital images with minimal distortion using *Artificial Neural Networks*. The goal is to improve on current industry standards for lossy compression.

Application

Compress *CT Scans* used for the detection of cancer lesions. Better compression would improve patient record storage and access to remote healthcare, among other benefits.

Triple Bottom Line Analysis

Social

Enable CT scans to be more accessible for both patients and practitioners (i.e. Rural locations).

Economic[1]

Cost savings for hospitals in the form of storage costs.

Environmental

Reduced storage and transmission needs lead to reduced power consumption.

[1] John Halamka, “The Cost of Storing Patient Records” (2018)

Stakeholders

Patients[1,2]
Reduces number of misdiagnosis for safety.

Practitioners
Makes CT Scans available in remote/rural locations.

Technologists
Reduces processing time for technologists taking CT scans.

Healthcare System[3]
Reduces cost of storage for CT scans.

[1] Singh. S, “Effect of CT Quality on Lung Module Detection and Management”, Journal of the American College of Radiology (2014)
[2] Smith-Bindman R. “Radiation Dose Associated with Common CT Examinations and Associated Lifetime. Risk of Cancer”, JAMA (2013)
[3] J. Halamka, “The Cost of Storing Patient Records” , GeekDoctor (2011)

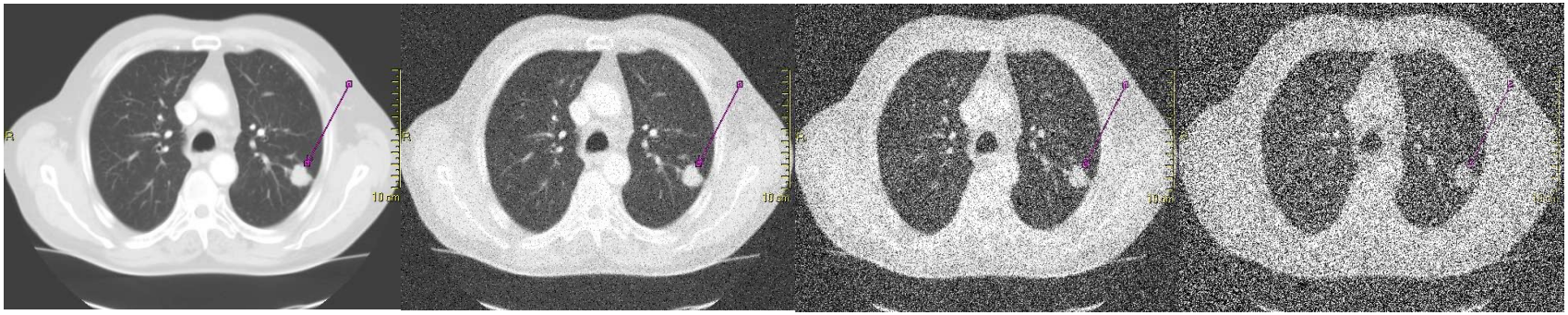
Data Compression

Lossy data compression aims to minimize the expected distortion between the original and reconstructed source

$$D(Q) = E[d(X, Q(X))]$$

for a distance metric d [1]

Distortion

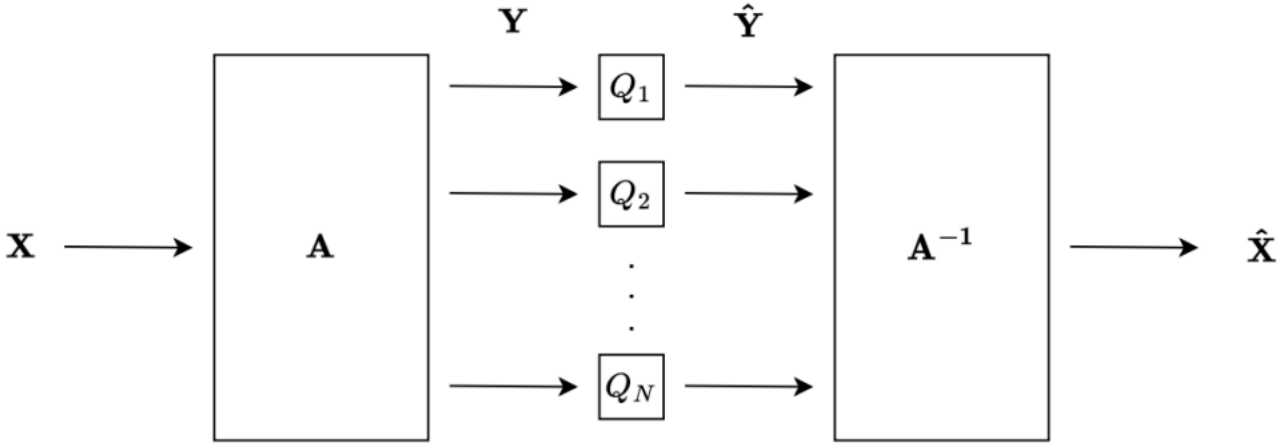


Compression Rate

[1] J. Cheng, "Channel Optimized Quantization of Images Over Binary Channels with Memory," Queen's University Masters thesis (1997)

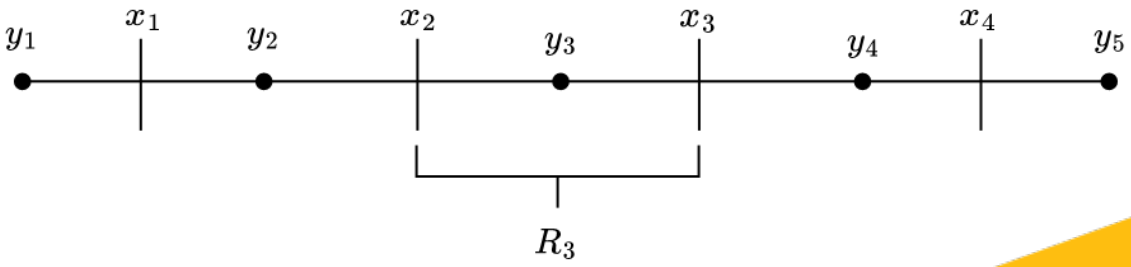
Image Source: "NIH Clinical Center Releases Dataset of 32,000 CT Images", NIH (2018)

Linear Transform Coding



- Uses **orthogonal** transformation A, *i.e.* $\mathbf{A^T = A^{-1}}$
- Scalar quantizer Q_i is the map $Q_i : \mathbb{R} \rightarrow \mathcal{C}_i$ for codebook

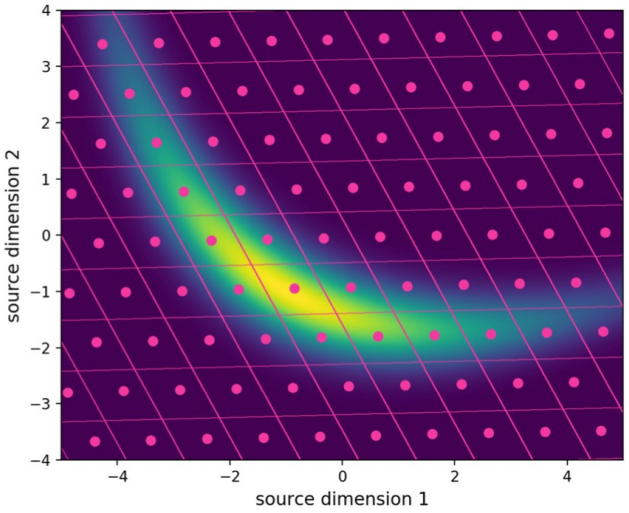
$$\mathcal{C}_i = \{y_1, \ldots, y_N\} \subset \mathbb{R}$$



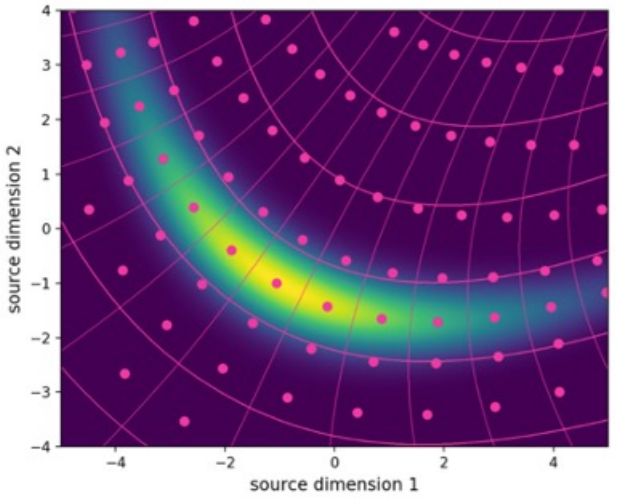
[1] J. Cheng, "Channel Optimized Quantization of Images Over Binary Channels with Memory," Queen's University Masters thesis (1997)

Motivation

- Linear transform coding maps the source to latent space prior to quantization
- Orthogonal matrix transforms impose linear structure on quantized bins in latent space, and Standard Non-Linear Transforms were previously not computationally feasible in higher dimensions
- Neural networks can arbitrarily approximate any continuous function



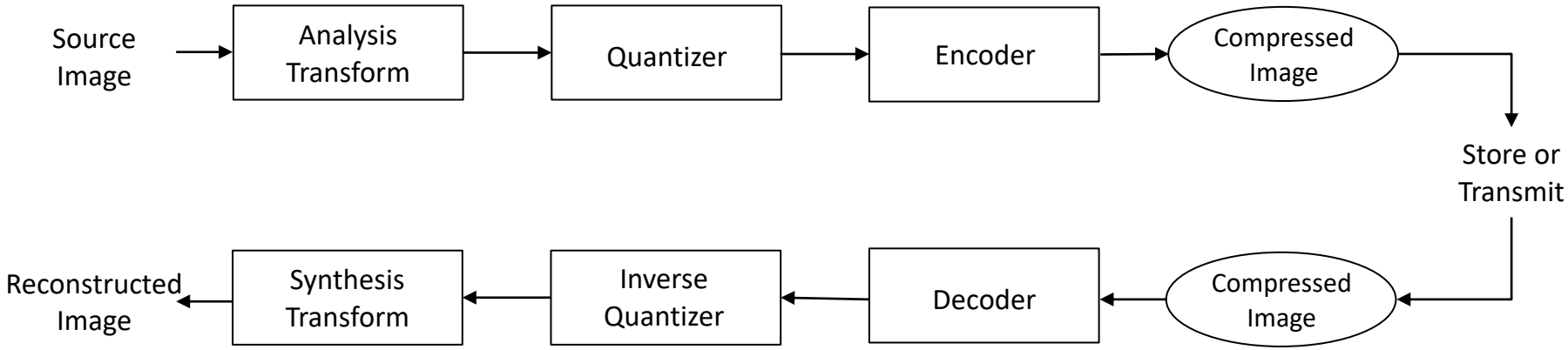
Linear Transform Coding
quantizer bins



Nonlinear Transform Coding
quantizer bins

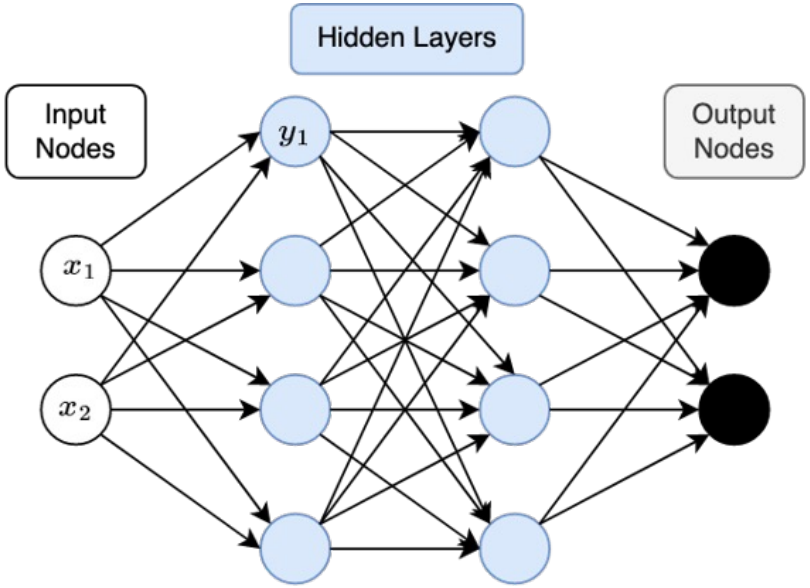
Image source: J. Ballé, “Nonlinear Transform Coding” (2021)

Solution – Overview



Goal: Implement Neural Network-based Analysis and Synthesis Transforms

System Design



Perform **Stochastic Gradient Descent** on cost function

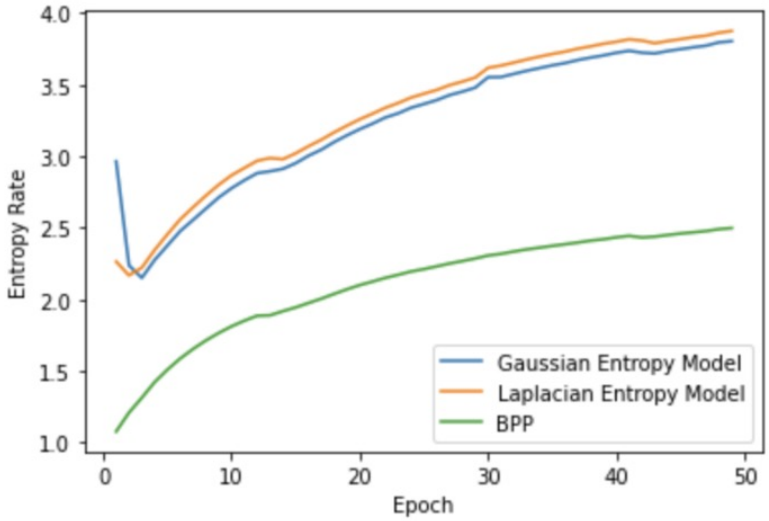
$$L_{NTC} = \mathbb{E}_x[-\log(P(\lfloor g_a(\boldsymbol{x}) \rceil) + \lambda d(\boldsymbol{x}, g_s(\lfloor g_a(\boldsymbol{x}) \rceil))]^{1}$$

for analysis transform $g_a(\boldsymbol{x})$, synthesis transform $g_s(\boldsymbol{x})$, P an entropy model of the quantized latent vector, and λ used for Lagrangian Optimization

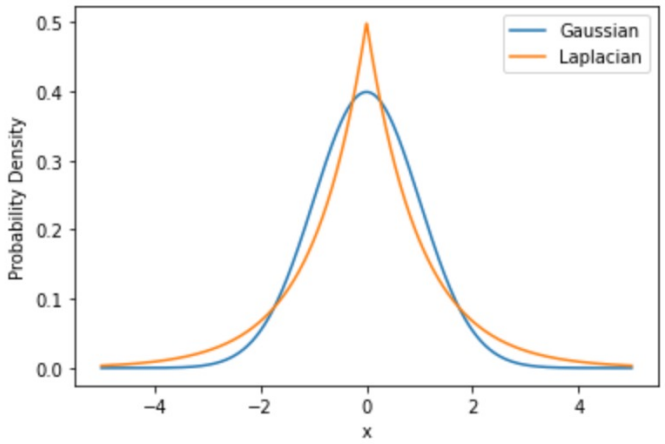
[1] J. Ballé, "Nonlinear Transform Coding" (2021)

Entropy Modelling

- Must choose a continuous entropy model to estimate and minimize the rate of the quantized latent vector
- Common choices are fixed-mode entropy models, including the Gaussian and Laplacian Distributions[1]



Estimated entropy rates and bit rates of images during model training

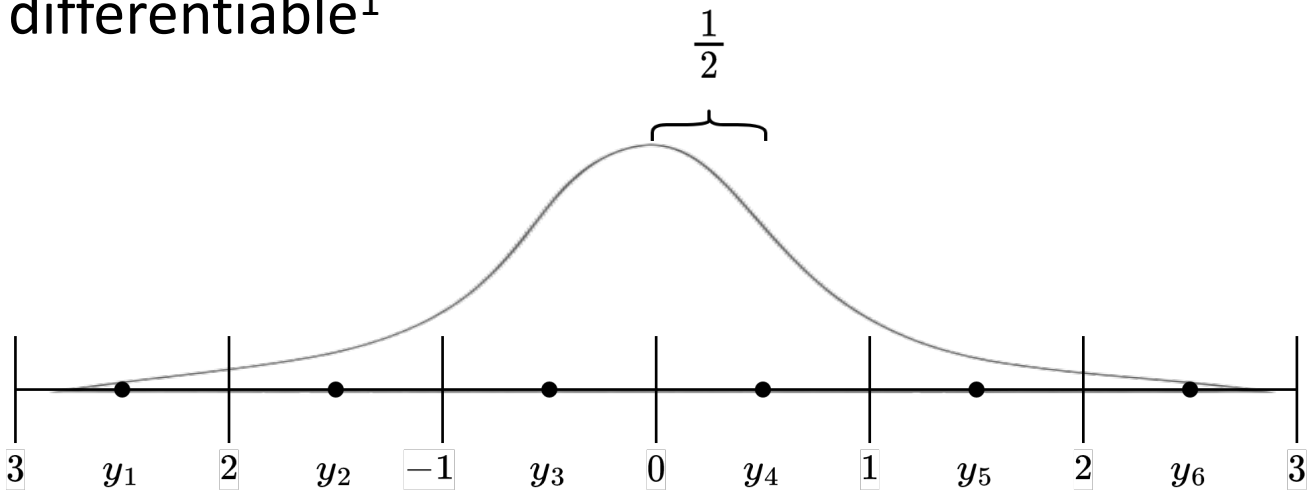


Normal and Laplacian Distributions

[1] J. Ballé, "Nonlinear Transform Coding" (2021)

Uniform Noise Perturbations

- Quantization approximated as Uniform$\left(\frac{-1}{2}, \frac{1}{2}\right)$ noise to make L_{NTC} differentiable[1]



- Cost function becomes

$$L_{NTC} = \mathbb{E}_x[-\log(P(g_a(\boldsymbol{x}) + \Delta) + \lambda d(\boldsymbol{x}, g_s(g_a(\boldsymbol{x}) + \Delta))]$$

[1] J. Ballé, "Nonlinear Transform Coding" (2021)

Neural Network Architecture

- **Convolution Layers**: Capture the spatial relationships and local patterns present in image data
- **Max Pooling Layers**: Reduce spatial dimensions (down-sample) and aids in translation invariance and robustness to noise
- **Fully Connected Layer**: Maps the image to a lower-dimension latent space for quantization (compression)

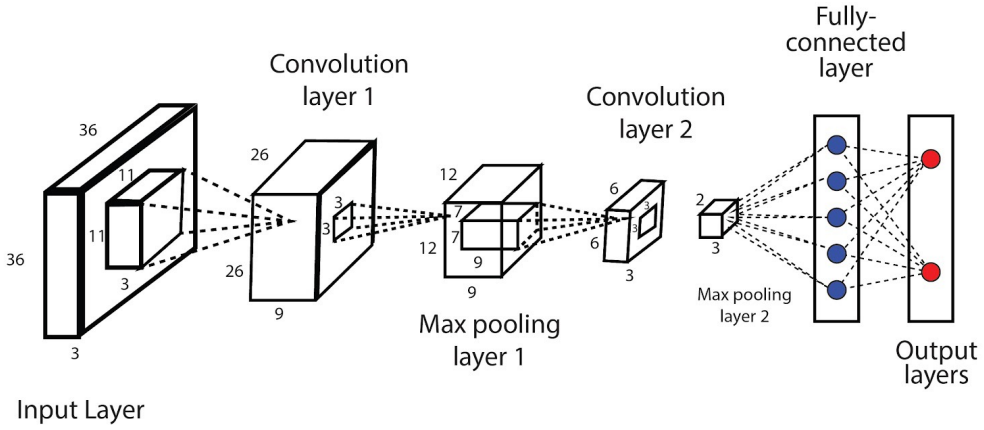
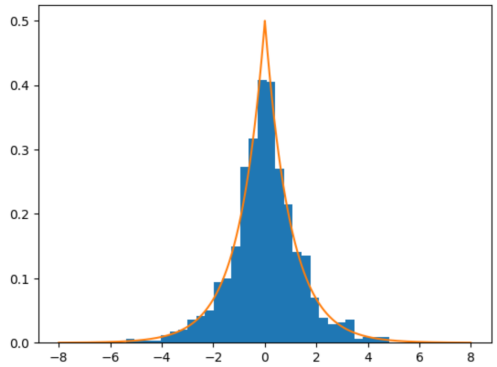
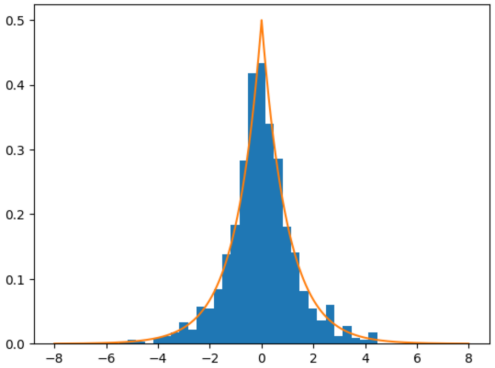


Image Source: Tanesh Balodi, "Convolutional Neural Network (CNN)", AnalyticSteps (2021)

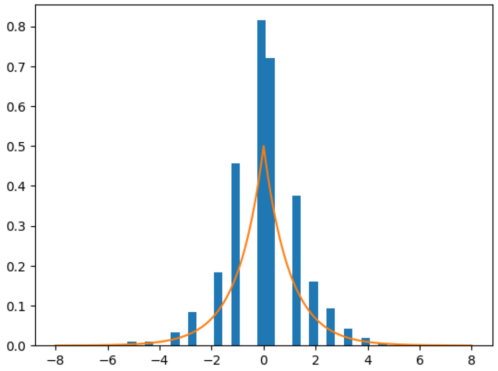
Model Testing – 1-D Laplacian Distribution



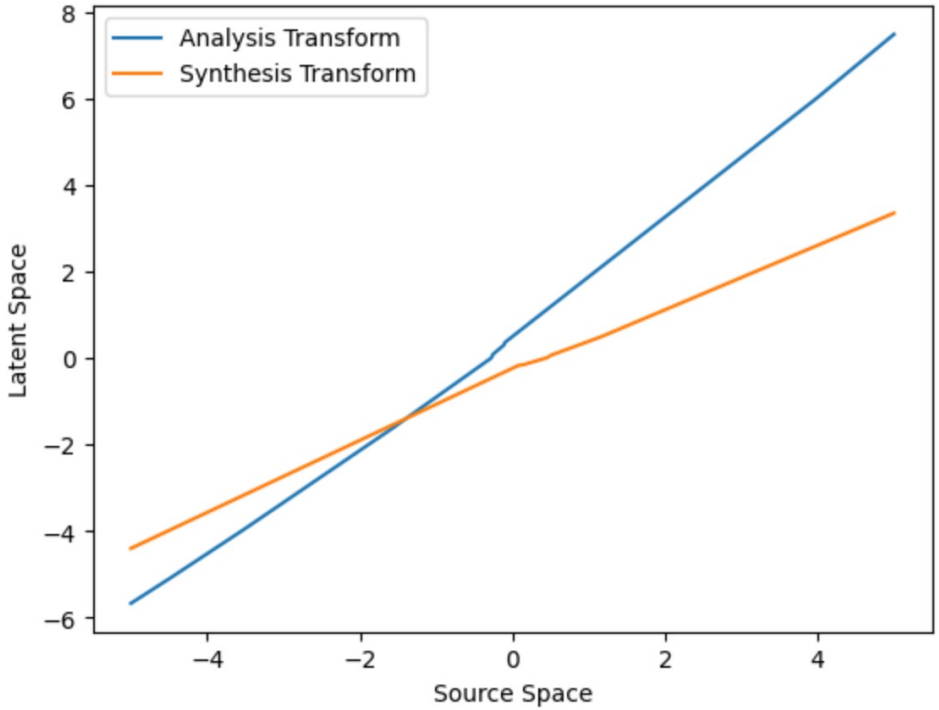
Original distribution



Reconstructed distribution using perturbations



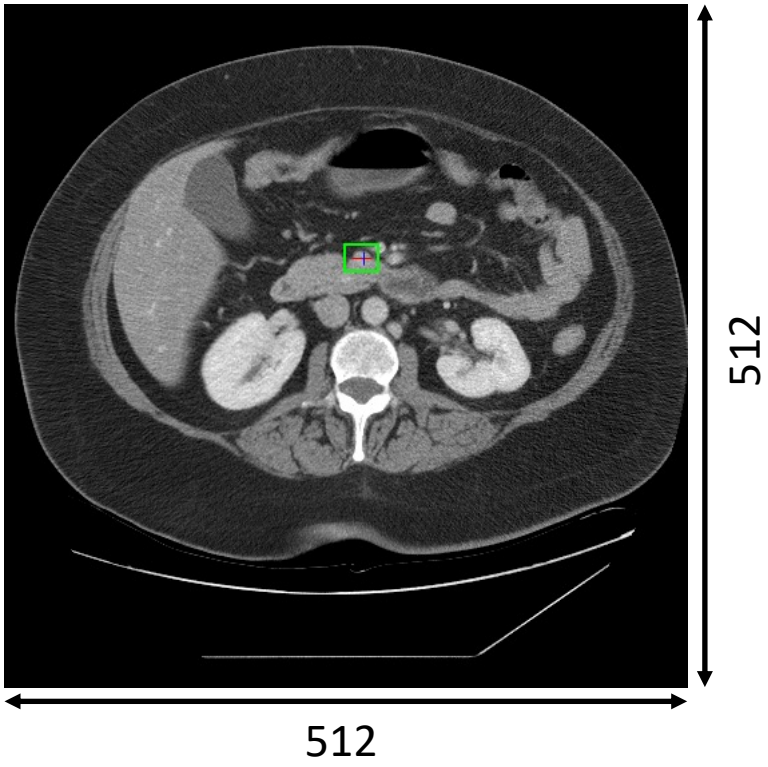
Reconstructed distribution using Quantizers



Analysis and synthesis transforms visualized in 2-D

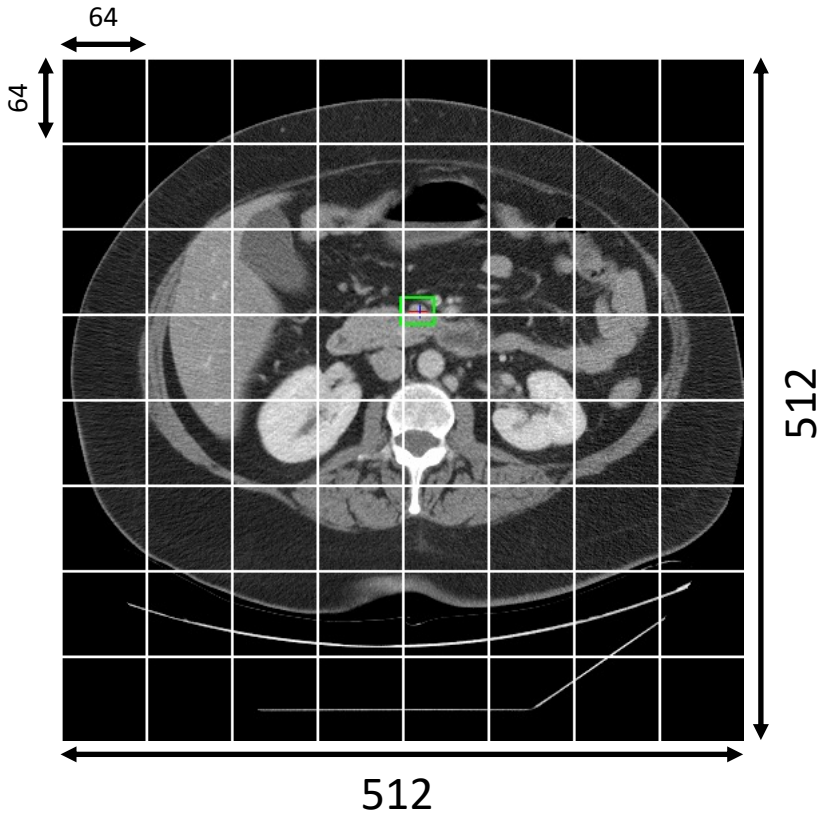
Model Iterations – Blocks

- Large image sizes can be **computationally impractical** for neural network training.
- The number of trainable parameters **scales quadratically with the input dimensions**
- Our 64x64 block model has **4.5 million** trainable parameters to optimize



Model Iterations – Blocks

- Large image sizes can be **computationally impractical** for neural network training.
- The number of trainable parameters **scales quadratically with the input dimensions**
- Our 64x64 block model has **4.5 million** trainable parameters to optimize



Model Iterations – SSIM Distortion Measure

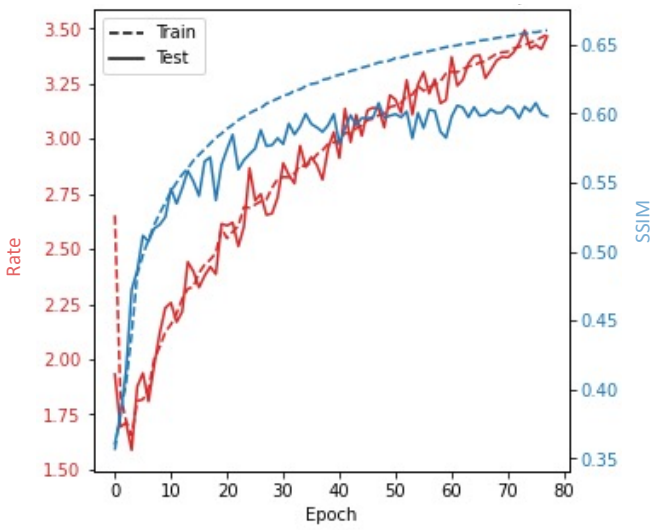
- **Problem**: MSE only measures difference between pixel values and does not consider the spatial arrangement or perceptual differences between images.
- **Structural Similarity Index Measure (SSIM):** A method to evaluate the similarity between two images that corresponds with human perception of image quality

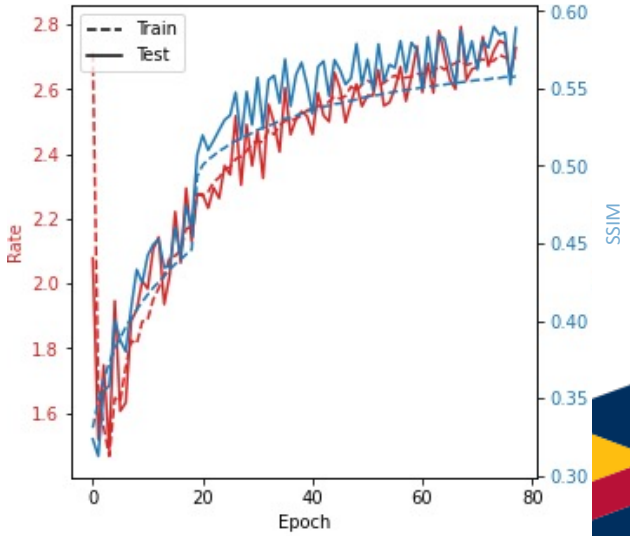
Image Source: P. Ndajah , "SSIM Image Quality Metric for Denoised Images", WSEAS (2010)

Model Iterations - Overfitting

- **Overfitting**: When a model becomes overly complex and fits the training data too closely, leading to poor performance on new, unseen data.

- **Solution**:
  - Dropout layers: Randomly drop neurons during training
  - Weight decay: Penalize large weights
  - Diverse training set: expose the model to wider range of images





Model Iterations – Activation Functions

- **Activation Function**: A function that determines the output of a neural network node based on its input
- **Rectified Linear Unit (ReLU)**: Introduces non-linearity and helps the network to learn more complex features.
- **Generalized Divisive Normalization (GDN)**: Improves the stability and robustness of the network by normalizing outputs to have zero mean and unit variance[1]

$$\mathrm{ReLU}(x) = \max(x, 0)$$

$$\mathrm{GDN}(x) = \frac{x}{\sqrt{\beta + \sum_j (x_j^2 \cdot \omega_j)}}$$

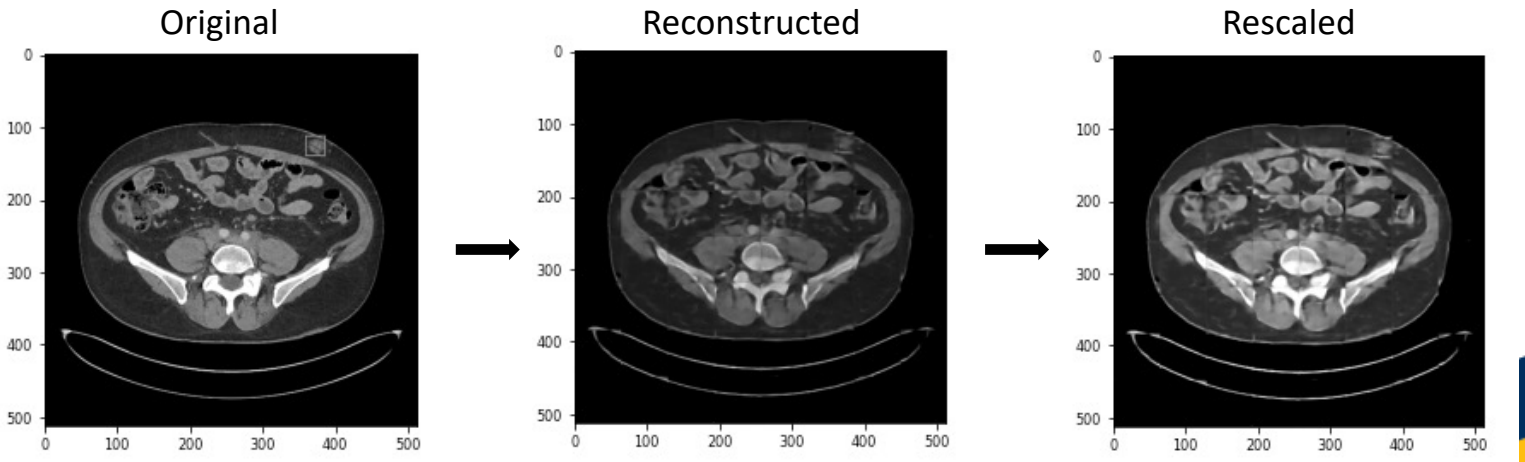
[1] J. Ballé, Efficient Nonlinear Transforms for Lossy Image Compression (2018)

Model Iterations – Contrast Intensity Rescaling

- **Problem**: Activation functions can compress the dynamic range of the image pixels resulting in loss of contrast[1]
- **Contrast Rescaling**:

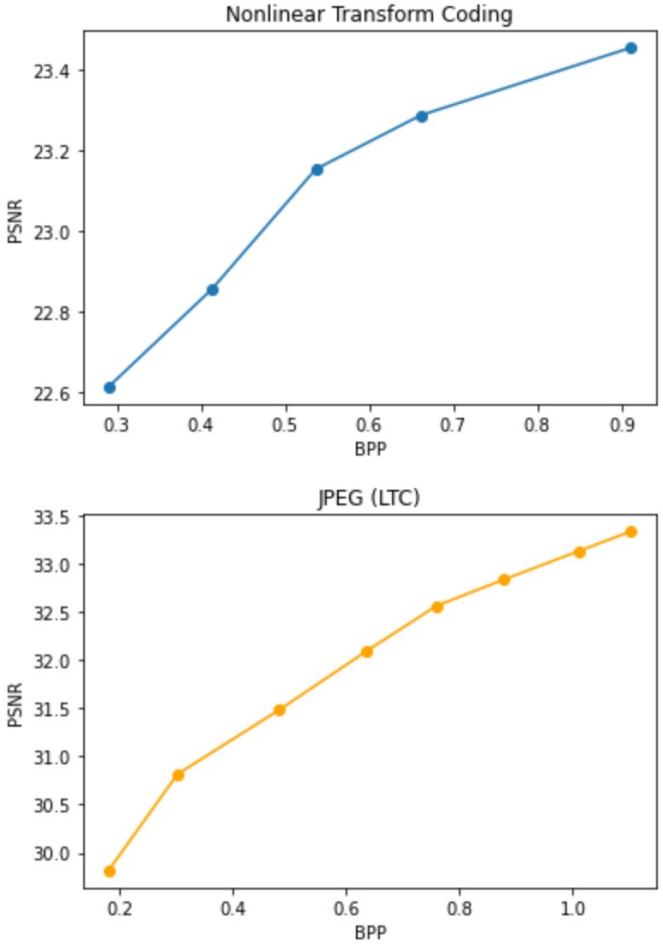
  Let $I_{i,j}$ be the intensity value of the pixel located at (i, j). Further, let p_1 and p_{99} represent the 1st and 99th percentiles of the intensity distribution of the original image. Then we can rescale the reconstructed image as follows:

$$I_{\text{rescaled}}(i,j) = \frac{I_{i,j} - I_{\min}}{I_{\max} - I_{\min}} \cdot (p_{99} - p_1) + p_1$$



[1] Y. Shik Moon, "Low Contrast Image Enhancement Using Convolutional Neural Network", ASTES (2019)

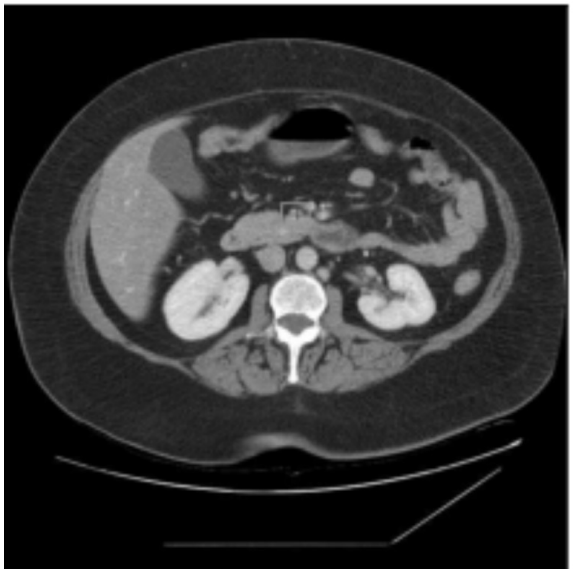
Quantitative Results



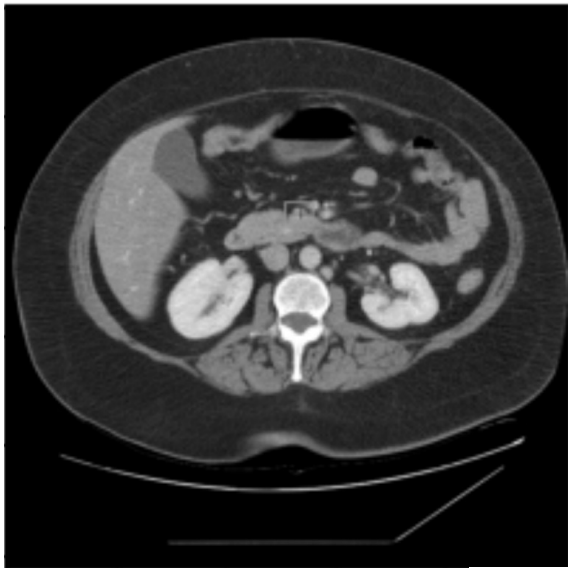
Rate Distortion graphs of our method and JPEG on the CT Scan dataset

- **Note:** Our method is trained on and optimized for *SSIM*
- **Comparison:** JPEG significantly outperforms our system using the test distortion measure of Peak Signal-to-Noise Ratio (PSNR)
- $PSNR = 10 \log_{10} \frac{255^2}{MSE}$

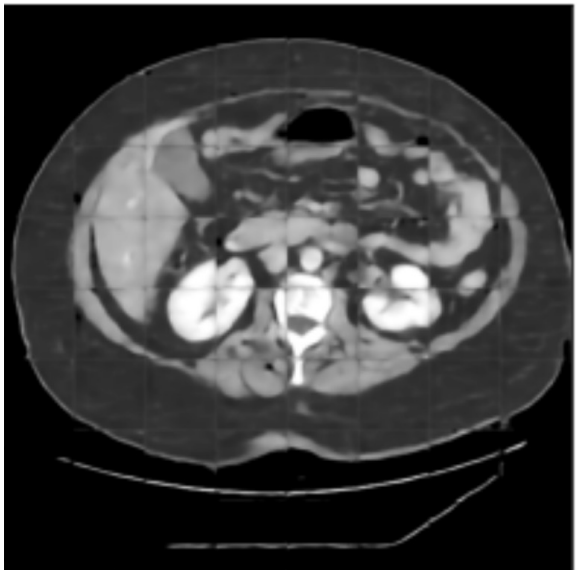
Qualitative Results



JPEG (0.77 BPP)



Original



Our Method (0.77 BPP)

Limitations and Possible Improvements

Entropy Model

Use mixture distributions that fit to the data or models.

Computational Resources

More computing resources → Train the model on whole image.

Testing Time

Tune hyperparameters and model architecture.

Entropy Rate

Use techniques such as Huffman or arithmetic coding.