Improving Image Compression and Restoration Process Using Neural Networks

Jia Zhuang
jzhuang7@stanford.edu
EE367 Project Proposal

I. Introduction

Traditional image compression and restoration process have been using variety of algorithms to minimize pixel-wise losses and to optimize MSE and PSNR (peak signal-to-noise ratio) characteristics. However, the resulting images usually can still be blurred and the high spatial frequency components are not well reconstructed.

Convolutional Neural networks (CNN) have been widely implemented in many areas of computer vision and image processing to improve the quality of the images and videos. Many literature have reported to produce a better image quality compressed and restored with artificial neural network algorithms, like autoencoder and/or GAN. Image compression process utilizing Convolutional Neural Networks (CNN) can outperform traditional compression techniques and denosing post-process methods [1].

From precious studies, many literature has focused on tuning the architecture of the network for specific goals. However, with different architectures in CNN, the loss layer is equally important during restoration process since it serves as a learning target that drives the network learning. Therefore, this project is divided into two sections: first a typical CNN and a generative compression model will be constructed and compared to reveal the advantages and disadvantages of paring autoencoder with GAN. Secondly, different lose functions will be compared using one type of neural network and methods to quantify and evaluate image quality and similarity will be investigated [2].

II. Previous Literature and Related Work

A. Generative Compression

The lossy image compression was traditionally formulated as a rate-distortion optimization problem. In Generative compression model, the process is framed to include an analysis transform, which maps the image data to a latent code space and a synthesis transform, which maps back to original space, and the compression is achieved by entropy encoding/decoding. During synthesis transform of the data, a simplified DCGAN is adopted to the process in order to achieve higher-quality images. The generative compression has demonstrated orders of magnitude improvement in both compression factor and noise tolerance compared to traditional techniques [3].

B. Loss Functions for Image Restoration

In the image restoration stage with neural network, the loss layer is a critical target for network learning. In many cases, the cost function is defaulted to be the squared L_2 norm, which is a widely understood error metric. However, in human vision systems, the L_2 norm can be naive with the fact that the sensitivity of noise depends heavily on luminance, contrast and structure. Alternative error metrics have been evaluated and compared again L_2 including L_1, Structural Similarity Index ($(MMIS)$) and Multi-scale structural Similarity Index ($MS-SSIM$) [4].

III. Project Overview and Approach

In this project, convolutional neural networks (for instance convolutional auto-encoder(CAE) and generative compression model) will be considered and compared for both compression and restoration process and various error metrics and loss functions will be implemented.

A. Neural Network Architectures

(1).png

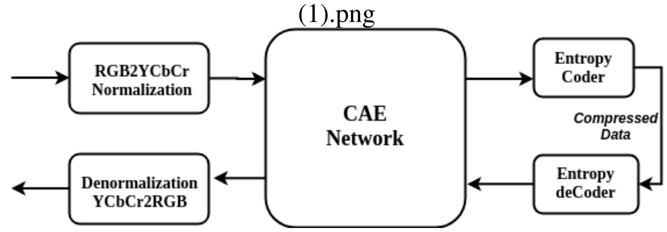

Fig. 1: Block diagram of CAE based image compression [5]

Convolutional Autoencoder, one type of CNN is illustrated in the figure 1. A typical CAE includes pre-processing (convert image data to read-in data), CAE comptation, PCA rotation and entrpy coder/decoder [1].

The generative compression architecture is built by pairing the encoder/decoder interface of VAE with GAN to take advantage of the high-quality images that are usually produced by GAN. As shown in figure 2, the structure has two major components: a decoder network, which is implemented by DCGAN and encoder network, which is trained to minimize

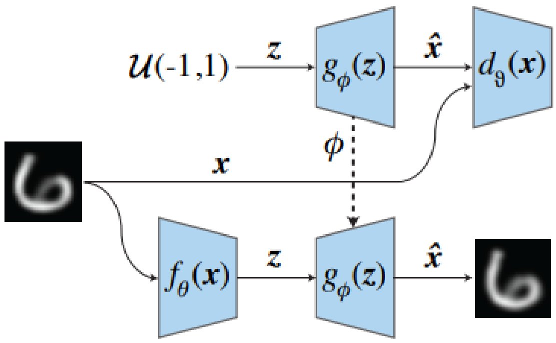

Fig. 2: Block diagram for Generative compression [3]

the distortion loss. The cascade of GAN and autoencoder improves the robustness of the neural nets and produces perceptually better restored images compared to previous end-to-end frameworks [4].

B. Loss Functions

Except for the L_2 error metric, following error metrics and loss functions will be considered and implemented in the neural network to see how the image produced differ.

The L_1 error metric: L_1 loss functions minimizes absolute differences between estimated values and target values, so L_1 does not over-penalize larger errors.

$$L_(L_1)(P) = \frac{1}{N} \sum_{p \in P} \mid (p) - y(p) \mid \tag{1}$$

where p is the index of pixel and P is the patch. $x(p)$ is the estimated value and $y(p)$ is the reference value.

SSIM: SSIM index is perceptually motivated and it quantifies the image degradation caused by data loss during compression process. SSIM calculation requires to look at not only the current pixel of interest but also the neighboring pixels.

$$SSIM(p) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \times \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{2}$$

Loss function for SSIM:

$$L_{SSIM}(P) = \frac{1}{N} \sum_{p \in P} 1 - SSIM(p) \tag{3}$$

MS-SSIM:

$$MS - SSIM(p) = l_M^{\alpha}(p) \times \prod_{j=1}^{M} cd_j^{\beta_\beta}(p) \tag{4}$$

where $l_M = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$ and $cs_j = \frac{2\sigma_{xy} + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$. Similarly, we can define the loss function for MS-SSIM as following:

$$L_{MS-SSIM}(P) = \frac{1}{N} \sum_{p \in P} 1 - MS - SSIMp \tag{5}$$

IV. Milestones

Stage 1: Compare neural network structures on compression and restoration of images. More specifically, use convolutional auto-encoder and generative model (autoencoder and GAN pair).

Stage 2: Implement various loss functions in the loss layer of the neural network, which compares the output result with the reference.

Timeline: Each stage will approximately take one to two week, and total of four weeks will be spent on this project.

Goal: The core objective of this project is to improve the image compression process with a cascade neural network structure and/or choosing and optimizing the loss function.

References

[1] W. Tao, F. Jiang, S. Zhang, J. Ren, W. Shi, W. Zuo, X. Guo, and D. Zhao, "An end-to-end compression framework based on convolutional neural networks," in *2017 Data Compression Conference (DCC)*, April 2017, pp. 463–463.

[2] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," *arXiv preprint arXiv:1703.00395*, 2017.

[3] S. Santurkar, D. Budden, and N. Shavit, "Generative compression," *2018 Picture Coding Symposiums*, pp. 258–262, 2018.

[4] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, March 2017.

[5] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Deep convolutional autoencoder-based lossy image compression," 06 2018, pp. 253–257.