Optimization of Fourier Space Filter for Chest X-ray Image Compression in Deep Learning Training

Xinyi Li, Yuhao Yan
BME548L Machine Learning and Imaging
xl233@duke.edu, yy263@duke.edu

Abstract

Fourier Transform (FT) compression using Fourier space filter was investigated for its effect on deep-learning-based medical image classification. A trainable filter was set with an initial shape as Butterworth filter, and optimized during the training process. Normal compression adopting a bilinear interpolation was also investigated and served as baseline. A published neural network architecture, ResNet18, was modified and used across all the training for consistency. Results showed that the optimized FT compression obtained a slightly higher classification accuracy compared to normal compression. More tests are needed for statistics significance. The optimization of the low-pass filter in FT compression was not significant.

1. Introduction

The use of medical imaging such as x-ray, CT and MRI increases dramatically in recent decades. The boosting volume of these digital images poses serious challenges to the electronic health record system regarding data storage and transmission. Therefore, medical images are often compressed for storage and access convenience. A desired compression technique should be able to reduce the file size while maintaining a comparable visual quality and keeping most information. Many compression methods have been proposed to fulfill this purpose, including wavelet-base compression, single value decomposition (SVD), etc. [1]

In recent decades, Deep Learning (DL) has become a robust tool for many clinical tasks in medical imaging, such as classification, segmentation, and reconstruction. As medical image databases become popular in DL related studies, medical image compression obtained new purposes and objectives. DL network training relies on high-performance GPU, which implies that computation time and memory are limited. Large input sizes or a large number of trainable parameters could consume enormous amount of computation resources. Therefore, in our understanding, an ideal image compression algorithm for AI should minimize storages without losing critical information that affects network performance.

Based on this understanding, we designed a network training experiment to explore the role of Fourier Transform (FT) in image compression algorithm. An open-source dataset contains thousands of Chest X-ray images was employed to mimic a clinical background. [2] The proposed image compression algorithm starts from cropping in the Fourier space. Next, a trainable filter that was initialized with Butterworth filter was applied to the image in Fourier space. Finally, the image was transformed back to image space through inverse Fourier transform and normalized for the following network calculations.

2. Method

General workflow of this work is shown in Figure 1. The raw data was processed to generate training and test data. Two different physical layers, normal compression and FT compression, were added to a published neural network architecture and got trained, respectively. The trained models were then evaluated for classification accuracy and the performance was compared. Details are explained in the following sections.

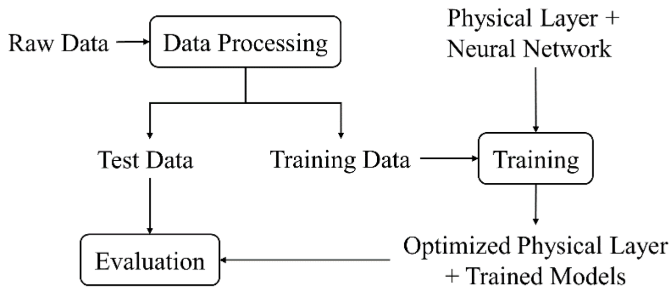

Figure 1: General workflow

2.1. Raw data

The raw data was acquired from the public database, Chest X-Ray Images (Pneumonia). [2] It was originally proposed for the purpose of deep-learning-based classification and referral of human diseases. It contains a total of 5863 chest x-ray images, each categorized into either normal or pneumonia image. All images are in .jpeg format and nicely organized into three folders including training, test and validation. Image sizes vary from file to file but are all in landscape view. Examples are shown in figure 2.

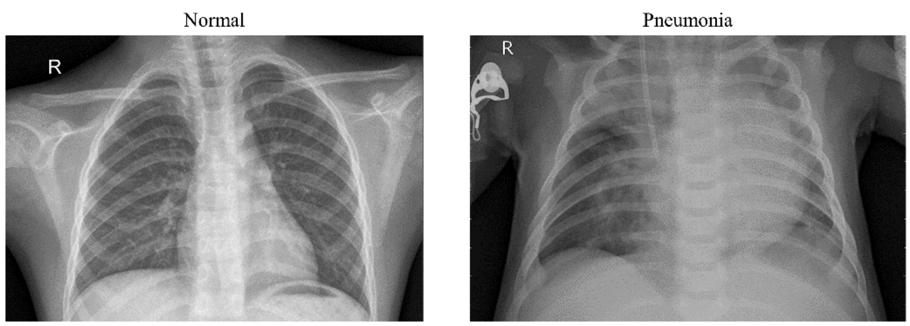

Figure 2: Examples of raw chest x-ray images. Left: Normal. Right: Pneumonia

2.2. Data processing

Raw data was firstly processed for consistency and simplicity. All the images were centrally cropped into a square matrix, resized into 256×256 and normalized. The normalized 256×256 square images were designated as true information. The training dataset was consisted of 500 normal images and 500 pneumonia images randomly picked from the raw database. The test dataset included 100 normal images and 100 pneumonia images, also randomly picked from the raw database and excluded from the training dataset.

Table 1: Dataset size

Category	Training	Test
Normal	500	100
Pneumonia	500	100

2.3. Physical layer: Compression

Two compression methods were introduced and compared in this work. Both compression methods aim to compress the 256×256 image to 64×64, so that the size of the file could be reduced by a factor of 16.

The first compression method was normal compression, which was a fixed method and served as a baseline. It was realized by bilinear interpolation of neighboring 16 pixels followed by normalization. [3]

The second one was Fourier Transformation (FT) compression, which was to be optimized in this work. The compression process was mimicked as follows. The image data was firstly Fourier transformed to frequency space data, then centrally cropped into a 64×64 matrix to satisfy the discrete Nyquist sampling theorem.

Next, a trainable filter was applied to the frequency space data to mimic the low-pass filter during image acquisition. Hereby we selected Butterworth filter as the initial low-pass filter. Butterworth filter can be expressed as [4]

$$F(\omega) = \frac{1}{\sqrt{1 + (s \times \omega)^n}} \tag{1}$$

Where ω is the frequency, s is the scaling factor, and n is the order. In this work, the initial Butterworth filter was set with a scaling factor of 0.03 and an order of 12. The 64×64 filter matrix was allowed to be trained. Finally, an inverse Fourier transformation was applied, followed by normalization, to get the compressed image.

2.4. Neural network

A published neural network, ResNet 18, was used in this work. [5] Some modifications were made for computation power consideration. The input size was adapted to 64×64. 5 residual blocks were removed, and all filter numbers were reduced by a factor of 8. Output was adapted to a single output with sigmoid activation. The code was acquired from the open source [6].

2.5. Training

The training parameters were set the same for all the training as follows. Training loss was defined as binary cross-entropy. Optimizer was set as Adam method with a learning rate of $1e-5$. Monitor loss was defined as mean, and the monitor accuracy was set as binary accuracy with a threshold of 0.5. Both models were trained for 20 epochs. The training was completed using Google Colab@.

2.6. Evaluation

The receiver operating characteristic curve (ROC) was generated, and the area under curve (AUC) was calculated to quantitatively compare the performance of classification. Qualitative evaluation was adopted to check the quality of compressed image and optimization of low-pass filter.

3. Result

The loss and accuracy for training and test are shown in figure 3 for demonstration. The loss tended to converge around 20 epochs.

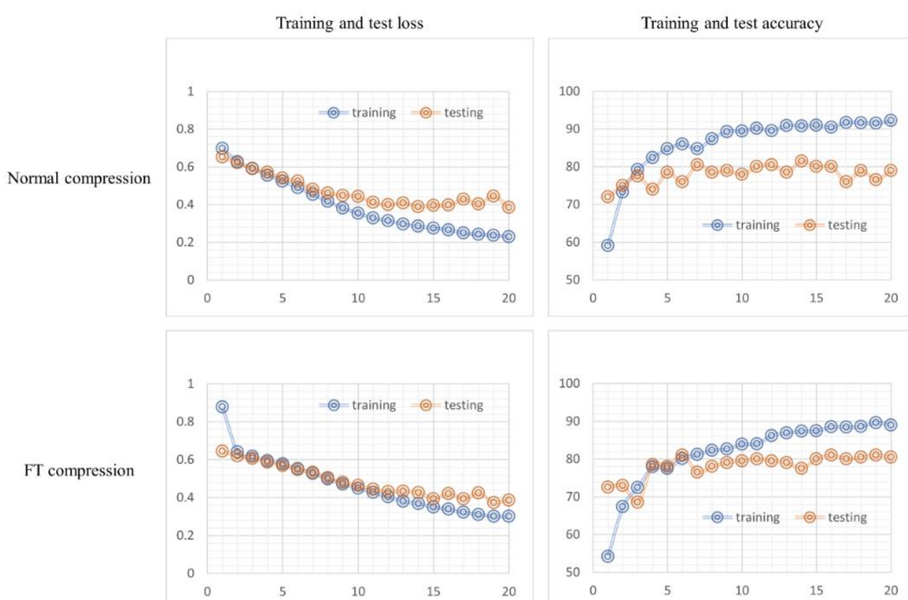

Figure 3: Training and test loss and accuracy. Left column: Loss. Right column: Accuracy. Top row: Normal compression. Bottom row: Fourier transform (FT) compression

The ROC was furtherly calculated and shown in figure 4. The AUC for the classification using normal compression was calculated as 91.61, while the AUC for the one using FT compression after optimization slightly improved to 91.92.

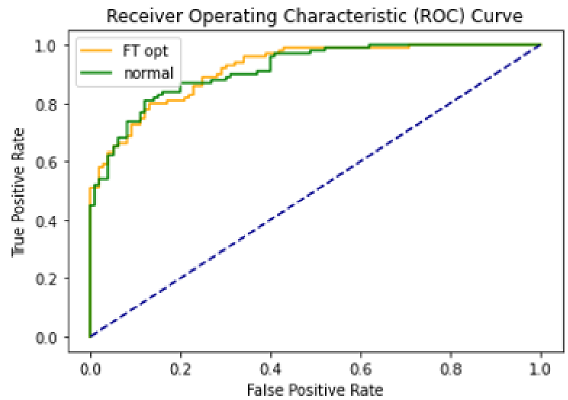

Figure 4: Receiver Operating Characteristic (ROC) curve. AUC of normal compression=91.61. AUC of FT compression after optimization =91.92.

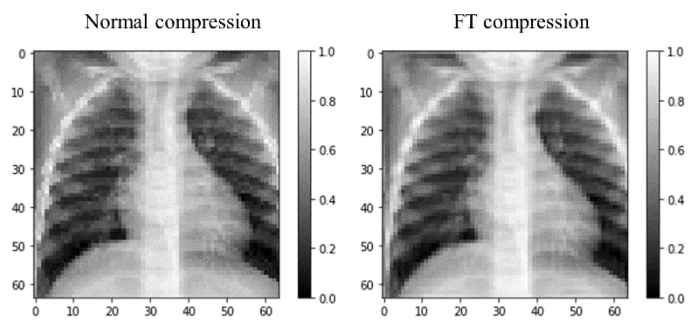

Figure 5: Image after compression. No significant difference is observed.

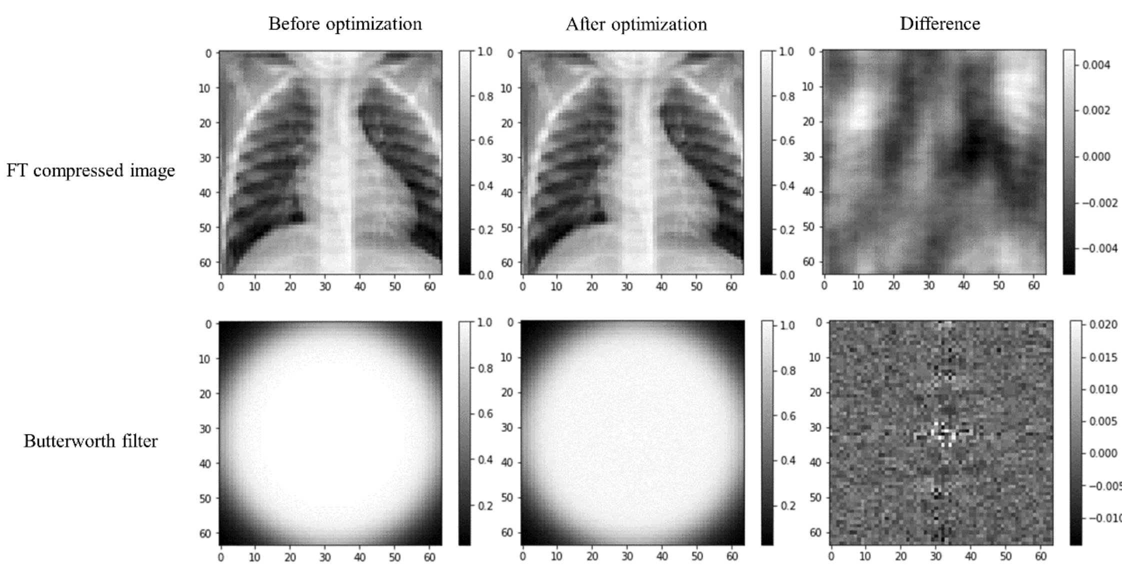

Figure 6: FT compressed image and Butterworth filter optimization. Differences are relatively insignificant.

By qualitative analysis, the results of a designated image were shown to compare the compression image quality and demonstrate filter optimization. Figure 5 shows the differences between two compressed images adopting normal compression and FT compression, respectively. No significant difference is observed. Figure 6 shows the compressed images and Butterworth filters before and after optimization. Noting that the magnitude of the differences is relatively insignificant for both the compressed images and the Butterworth filters.

4. Discussion

The outcome shows that the classification performance of the selected neural network still outstands while applying image compression. This indicates that in some tasks image compression could be adopted to save the limited GPU memory and accelerate the training process.

The performance of classification adopting FT compression slightly improved compared to normal compression. However, the improvement was not very obvious, thus repeated tests are needed for statistics significance.

During the implement, it was observed that the initial values would greatly influence the training outcome. Starting point should be selected carefully, which poses challenges to adaption of the methodology in this work.

The Butterworth filter was optimized during the training process, which led to a better classification performance of the neural network. However, the physical meaning of the filter optimization is not clear and needs further investigation.

5. Conclusion

The classification performance of the selected neural network outstands while applying image compression. The developed Fourier transformation image compression filter could slightly improve the neural network training outcome compared to the normal bilinear interpolation compression.

Reference

[1] Kumar, Pardeep, and Ashish Parmar. "Versatile Approaches for Medical Image Compression: A Review." *Procedia Computer Science*, vol. 167, 2020, pp. 1380–1389., doi:10.1016/j.procs.2020.03.349.

[2] Kermany, Daniel; Zhang, Kang; Goldbaum, Michael, "Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Images for Classification", *Mendeley Data*, V2, 2018, doi: 10.17632/rscbjbr9sj.2.

[3] "Tf.image.resize: TensorFlow Core v2.4.1." *TensorFlow*, www.tensorflow.org/api_docs/python/tf/image/resize.

[4] Makandar, Aziz, and Bhagirathi Halalli. "Image Enhancement Techniques Using Highpass and Lowpass Filters." *International Journal of Computer Applications*, vol. 109, no. 14, 2015, pp. 21–27., doi:10.5120/19256-0999.

[5] Hashmi, Mohammad Farukh, et al. "Efficient Pneumonia Detection in Chest Xray Images Using Deep Transfer Learning." *Diagnostics*, vol. 10, no. 6, 2020, p. 417., doi:10.3390/diagnostics10060417.

[6] Kotikalapudi, Raghavendra. "Keras-resnet/resnet.py", Github repository, https://github.com/raghakot/keras-resnet/blob/master/resnet.py