Transfer Learning for Low-Dose CT Denoising

Hongming Shan, Yi Zhang, Qingsong Yang, Uwe Kruger, Wenxiang Cong and Ge Wang
Biomedical Imaging Center, CBIS/BME/SoE
Rensselaer Polytechnic Institute
SHANH@RPI.EDU

November 19, 2017
Low-Dose CT

- CT-associated high-dose x-ray radiation carries health risks for patients.
- Reduction of the radiation dose compromises CT image quality, and the resultant image noise can compromise diagnostic information.

Quarter-dose

Full-dose

Images are from 2016 NIH-AAPM-Mayo Clinic Low-Dose CT Grand Challenge
Noise Reduction for Low-Dose CT

• **Sinogram filtration**
  • Perform on either raw data or log-transformed data

• **Iterative reconstruction**
  • Optimize an objective function that combines the statistical properties of data in the sinogram domain and prior information in the image domain together

• **Post-processing techniques**
  • Operate on an image directly which has been reconstructed from raw data.
  • **Deep learning-based methods** achieving impressive results.
Deep Learning-based Denoising Method

- **Network architecture**: Complexity of model
  - Convolutional layer
  - Deconvolutional layer
  - Special connection

- **Objective function**: How to learn from image/data
  - Mean squared error (MSE), as well as L1 norm (Enhao’s talk)
  - Adversarial loss
  - Perceptual loss
## Network architecture

<table>
<thead>
<tr>
<th>Methods</th>
<th>Network architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Conv. Layer</td>
</tr>
<tr>
<td>CNN(^1)</td>
<td>√</td>
</tr>
<tr>
<td>RED-CNN(^2)</td>
<td>√</td>
</tr>
<tr>
<td>GAN-3D(^3)</td>
<td>√</td>
</tr>
<tr>
<td>CNN-Cascade(^4)</td>
<td>√</td>
</tr>
<tr>
<td>WGAN-VGG(^5)</td>
<td>√</td>
</tr>
<tr>
<td>Ours</td>
<td>√</td>
</tr>
</tbody>
</table>

Traditional convolutional autoencoder includes convolutional layers and deconvolutional layers

- encoding low-dose CT image
- decoding to reconstruct normal-dose CT image
Contracting Path Convolutional Autoencoder (CPCA)

Contracting path copies the preceding feature maps and reuses them at later layers with the same feature-map sizes, preserving the details of the high resolution features.

- U-net\(^1\)
- DenseNet\(^2\)

**Objective function**

<table>
<thead>
<tr>
<th>Methods</th>
<th>Objective function</th>
<th>MSE</th>
<th>Adversarial Loss</th>
<th>Perceptual Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN$^1$</td>
<td></td>
<td>√</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RED-CNN$^2$</td>
<td></td>
<td>√</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GAN-3D$^3$</td>
<td></td>
<td>√</td>
<td>√</td>
<td>-</td>
</tr>
<tr>
<td>CNN-Cascade$^4$</td>
<td></td>
<td>√</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>WGAN-VGG$^5$</td>
<td></td>
<td>-</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Ours</td>
<td></td>
<td>-</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

**MSE**: Pixel-wise difference, Regression-to-Mean

**Adversarial loss**: Capture texture information, from same distribution, but samples are not matched very well

**Perceptual loss**: Measure similarity in feature space, parameters-fixed network
**Objective Function**

- **Adversarial loss**

\[
\mathcal{L}_a = \mathbb{E} \left[ D(I_{\text{est}}) \right] - \mathbb{E} \left[ D(I_{ND}) \right] + \lambda \mathbb{E} \left[ \left( \| \nabla D(\widetilde{I}) \|_2 - 1 \right)^2 \right]
\]

- **Perceptual loss**

\[
\mathcal{L}_p = \| \phi(I_{\text{est}}) - \phi(I_{ND}) \|^2_2.
\]

- **Objective function**

\[
\mathcal{L} = \mathcal{L}_a + \lambda_p \mathcal{L}_p.
\]
3D Denoising model

• Spatial information from adjacent LDCT slices
  ▪ Most of the existing denoising networks focus on image denoising in 2D.
  ▪ The adjacent image slices in a CT volume have strong correlative features that can potentially improve 2D-based image denoising.

• For example, we input one image with its 2 adjacent slices.
  ▪ **Input**: Augment one LDCT image with three LDCT images;
  ▪ **Filter**: Replace a $3 \times 3$ convolutional filter with a $3 \times 3 \times 3$ convolutional filter
Training 3D Denoising Model

Training from scratch?

Do transfer learning from a trained 2D model
2D filter to 3D filter

• We proposed a simple yet effective way to do transform from 2D filter to 3D filter

• Assume we have 2D filter $H \in \mathbb{R}^{3\times3}$, then corresponding 3D filter $B \in \mathbb{R}^{3\times3\times3}$ is

$$B(0) = 0_{3\times3}, \quad B(1) = H_{3\times3}, \quad B(2) = 0_{3\times3}.$$

• In this way, the 2D neural network and 3D neural network have same performance, then do fine-tuning to learn spatial information from adjacent slices.

• Spatial information is unknown for network, let it learn from data
  ▪ Suitable for any thickness in CT
Interpretation

• Under GAN framework, Generator $G$ and Discriminator $D$ are against each other.
  ▪ $D$ tells differences between fake samples and real samples
  ▪ $G$ fools $D$ by generating more similar samples
  ▪ $D$ depends on $G$
  ▪ $G$ depends on $D$

Balance between $G$ and $D$ is very important. Do not try to break it.
Experimental Data

• Experimental data from Mayo Clinic Low-Dose CT Grand Challenge
• Input: Quarter-dose CT images
• Output: Full-dose CT images
• Training data: 128K patches of size 64x64
• Validation data: 64K patches of size 64x64
Network Parameters

- No. of feature maps is 32 except for last layer which has only 1.
- Filter size: 3×3, stride is 1.
- ReLU is used after each convolutional layer.
- 1×1 convolutional layer is used to reduce number of feature maps from 64 to 32 after each contracting path.
- Hyperparameter $\lambda_p = 0.1$ via cross-validation
- Learning rate for training from scratch: $1.0 \times 10^{-4}$.
- Learning rate for transfer learning from 2D: $0.5 \times 10^{-4}$. (fine-tuning)
- Learning rate decays as epoch goes.
- Adam is used for optimization
Comparison: Training from Scratch

• CPCA-\(i\) denotes \(i\) slices are fed into CPCA.
  - \(i = 1\) : 2D NN
  - \(i = 3, 5, 7\) : 3D NN in our experiments.

• Validation results
Transfer Learning v.s. Training from Scratch

Transfer learning from a trained 2D model at epoch 10

Input: 3 slices

Transferred from this point
Transfer Learning v.s. Training from Scratch

Transfer learning from a trained 2D model at epoch 10

Input: 5 slices

Transferred from this point
Transfer Learning v.s. Training from Scratch

Transfer learning from a trained 2D model at epoch 10

Input: 7 slices

Transferred from this point
Comparison with State-of-the-Art

- Testing the trained denoising model on full-size CT image (1300 of size 512x512 in total)
- Comparing with recently published methods
  - REDCNN\(^1\)
  - WGAN-VGG\(^2\)

---

# Quantitative Analysis

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SSIM</th>
<th>Perceptual Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quarter-Dose</td>
<td>26.07</td>
<td>0.8340</td>
<td>4.81</td>
</tr>
<tr>
<td><strong>RED-CNN</strong></td>
<td>31.39</td>
<td>0.9194</td>
<td>4.31</td>
</tr>
<tr>
<td><strong>WGAN-VGG</strong></td>
<td>28.88</td>
<td>0.8957</td>
<td>2.55</td>
</tr>
<tr>
<td>CPCA-1</td>
<td>29.62</td>
<td>0.8976</td>
<td>2.37</td>
</tr>
<tr>
<td>CPCA-3</td>
<td>29.84</td>
<td>0.9004</td>
<td>2.06</td>
</tr>
<tr>
<td>CPCA-5</td>
<td>30.00</td>
<td>0.9023</td>
<td>1.99</td>
</tr>
<tr>
<td><strong>CPCA-7</strong></td>
<td><strong>30.01</strong></td>
<td><strong>0.9029</strong></td>
<td><strong>1.96</strong></td>
</tr>
</tbody>
</table>

RED-CNN: optimization using MSE loss leads to blurry output images due to regression-to-mean problem.
## Quantitative Analysis

<table>
<thead>
<tr>
<th>Method</th>
<th>PSNR</th>
<th>SSIM</th>
<th>Perceptual Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quarter-Dose</td>
<td>26.07</td>
<td>0.8340</td>
<td>4.81</td>
</tr>
<tr>
<td>RED-CNN</td>
<td>31.39</td>
<td>0.9194</td>
<td>4.31</td>
</tr>
<tr>
<td>WGAN-VGG</td>
<td>28.88</td>
<td>0.8957</td>
<td>2.55</td>
</tr>
<tr>
<td>CPCA-1</td>
<td>29.62</td>
<td>0.8976</td>
<td>2.37</td>
</tr>
<tr>
<td>CPCA-3</td>
<td>29.84</td>
<td>0.9004</td>
<td>2.06</td>
</tr>
<tr>
<td>CPCA_TF-3</td>
<td>30.00</td>
<td>0.9031</td>
<td>2.01</td>
</tr>
<tr>
<td>CPCA-5</td>
<td>30.00</td>
<td>0.9023</td>
<td>1.99</td>
</tr>
<tr>
<td>CPCA_TF-5</td>
<td>30.04</td>
<td>0.9032</td>
<td>1.90</td>
</tr>
<tr>
<td>CPCA-7</td>
<td>30.01</td>
<td>0.9029</td>
<td>1.96</td>
</tr>
<tr>
<td>CPCA_TF-7</td>
<td>30.14</td>
<td>0.9045</td>
<td>1.87</td>
</tr>
</tbody>
</table>
Case Study: [-180, 200]HU

Quarter-Dose
PSNR: 24.99
SSIM: 0.792
P.Los.: 5.33

RED-CNN
PSNR: 30.67
SSIM: 0.901
P.Los.: 4.76

WGAN-VGG
PSNR: 28.62
SSIM: 0.783
P.Los.: 2.76

Full-Dose
PSNR: 28.73
SSIM: 0.870
P.Los.: 2.43

CPCA-1
PSNR: 28.73
SSIM: 0.870
P.Los.: 2.43

CPCA_TF-7
PSNR: 29.20
SSIM: 0.878
P.Los.: 2.29
ROI: Metastasis

Quarter-Dose

RED-CNN

WGAN-VGG

Full-Dose

CPCA-1

CPCA_TF-7
Case Study: [-160, 240]HU

Quarter-Dose

PSNR: 22.82
SSIM: 0.799
P.Los.: 6.25

RED-CNN

PSNR: 28.28
SSIM: 0.886
P.Los.: 5.08

WGAN-VGG

PSNR: 26.28
SSIM: 0.863
P.Los.: 2.82

Full-Dose

PSNR: 26.67
SSIM: 0.867
P.Los.: 2.60

CPCA-1

PSNR: 27.12
SSIM: 0.872
P.Los.: 2.17

CPCA_TF-7
ROI: Metastasis

Quarter-Dose

RED-CNN

WGAN-VGG

Full-Dose

CPCA-1

CPCA_TF-7
Discussion

• How do curves look like if we initialize 3D filter using random initialization or closed-form extension from a trained 2D filter to a 3D counterpart based on symmetric consideration?

• What if the 2D model was not trained in the GAN framework?
  - Doesn’t matter. Train a discriminator from scratch to converge, then do transfer learning and fine-tuning.
Conclusion

• We have introduced contracting path convolutional autoencoder (CPCA) for low-dose CT denoising
• Optimized denoising model under WGAN framework
• Proposed a simple yet effective way of transfer learning from a 2D trained model to a 3D counterpart, avoiding 3D training from scratch
• Our work can be extended to higher dimensionality in other tomographic imaging scenarios
Future work

• Inspired by Prof. Quanzheng Li’s talk
  ▪ Evaluating denoising model via detection/classification task
Thank You!

Biomedical Imaging Center
Principal Investigator: Ge Wang

Wenxiang Cong  Qingsong Yang
Guang Li       Matthew Getzin
Hongming Shan  Lars Gjestebry
Ruibin Feng    Fenglei Fan
Tao Xu         Qing Lyu