Machine learning for large scientific images

D.M. Pelt
Computational Imaging Group, CWI

2019 Woudschoten Conference
Performed PhD research at CWI (2016)
  - Supervised by prof. dr. K. J. Batenburg
  - Topic: efficient tomographic reconstruction algorithms

After PhD spent a year as post-doc at Berkeley Lab
  - Supervised by prof. J. A. Sethian
  - Focus on developing machine learning algorithms

Currently post-doc at Computational Imaging group

Recently awarded NWO VENI grant (2018)
  - Title: Machine learning for large 3D tomographic images
Growing group of researchers focused on imaging
- Broad range of topics and expertises
- https://www.cwi.nl/research/groups/computational-imaging
Recent developments in ML for scientific images

- Brief introduction to machine learning for images
- Problems with popular existing approaches
- Recently proposed new approach for scientific images
- Results for various scientific problems
- Conclusions and outlook
Neural networks

- Model some unknown function \( f : \mathbb{R}^m \rightarrow \mathbb{R}^n \) using a nonlinear network \( g_\phi \) with parameters \( \phi \in \mathbb{R}^{N_\phi} \).

- \( f \) and \( \phi \) are unknown
Training networks

- **Training**: find $\phi$ such that $g_{\phi}(x) \approx f(x)$
- **Supervised learning**: pairs of $(x, f(x))$ are known

- **Partial derivatives** are easy and fast to compute
  - Backpropagate error function derivative using chain rule
Rapid increase in popularity after 2012
Even though math is much older!

2018 Turing award for Bengio, Hinton, and LeCun

Used daily by companies (Google, Facebook, etc)

Typically, use Convolutional Neural Networks (CNN)

CNN → cat!
- Learn mapping from input image \( x \) to output image \( y \)
- Each layer convolves images of previous layer
- Often small filters (e.g. 3x3), but many layers/images
- Learn filters by presenting input/target pairs
- Deep networks have many layers
- Typically, use **downscaling and upscaling** to capture features at different scales
- First encoder, then decoder
Applications of CNNs

Sources from GitHub: junyanz/CycleGAN, phillipi/pix2pix, pathak22/context-encoder
Potential for scientific images

- ML has proven **successful** for photographic images
- Has **large potential** for scientific images
  - For example, ML could improve image quality, enable automatic segmentation, perform analysis, …

How can machine learning be applied in science?
Obvious approach
Obvious approach
Problem 1: practical

- Complicated networks
  - How to choose which operations to use and how to combine them?

- Implementing the networks requires expert knowledge

- What works in one problem often does not work well for others
  - Changes are needed that require expert knowledge as well
Problem 2: mathematical

- Networks require **many parameters** to achieve accurate results (e.g. millions)
- Problem: networks prone to **overfitting** training set
  - Large networks are often pre-trained with a huge training set (e.g. ImageNet >10 million images)

- In scientific problems, training data is often **limited** (e.g. only a few images)
  - Training large networks with limited data is difficult
Problem 3: computational

- Networks require many layers and intermediate images to achieve accurate results
- Applying to large images requires prohibitively large computational costs
  - GPU memory, computation time

- Networks often tested on small images (e.g. 28x28 up to 256x256)
  - Typical synchrotron CT image is 2560x2560
New approaches specific to scientific images are needed!
Scaling operations impose structure.
Detected features have to be copied to deeper layers.
Decoder cannot be used to improve encoder.

Better: remove scaling, reuse features, mix decoder and encoder.
Solution 1: remove scaling

- Use *dilated convolutions* instead of scaling to capture features at larger scales

Solution 1: remove scaling
Solution 2: dense connections

- No scaling: all intermediate images are the same size
- Why limit input to previous layer only?
- In fact, can use all previous images as input!
  - Including input images
- To compute output, can use all intermediate images
Solution 2: dense connections

- No scaling: all intermediate images are same size
- Why limit input to previous layer only?
- In fact, can use all previous images as input!
  - Including input images
- To compute output, can use all intermediate images
Solution 3: mix scales

- Use **small and large dilations** throughout the network

- Result: encoder and decoder are effectively **mixed**
Mixed-scale dense network

- Dilated convolutions instead of scaling
- Each layer is a single image
- All previous images (including input) are used

(Pelt & Sethian, PNAS 2018)
Advantages

▪ Maximum reuse of features
  ▪ Detected feature is directly usable in deeper layers

▪ Fewer intermediate images are needed
  ▪ Larger images can be efficiently processed

▪ Fewer parameters are required
  ▪ Accurate training with few training examples

▪ Network can learn how to combine different dilations
  ▪ Same network can be applied to different problems
- 512x512 pixels
- 36 combinations
  - shape, size, texture
- Detect 6 combinations
## Results - road scene segmentation

<table>
<thead>
<tr>
<th>Method</th>
<th>Pars (M)</th>
<th>GA</th>
<th>CA</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS-D-Net (100 layers)</td>
<td>0.048</td>
<td>85.1</td>
<td>56.8</td>
</tr>
<tr>
<td>MS-D-Net (200 layers)</td>
<td>0.187</td>
<td>87.0</td>
<td>63.9</td>
</tr>
<tr>
<td>U-Net (3 scaling operations) (5)</td>
<td>1.863</td>
<td>83.2</td>
<td>50.4</td>
</tr>
<tr>
<td>U-Net (4 scaling operations) (5)</td>
<td>1.926</td>
<td>85.5</td>
<td>48.4</td>
</tr>
<tr>
<td>SegNet-Basic-EncoderAddition (4)</td>
<td>1.425</td>
<td>84.2</td>
<td>56.5</td>
</tr>
<tr>
<td>SegNet-Basic (4)</td>
<td>1.425</td>
<td>84.0</td>
<td>54.6</td>
</tr>
<tr>
<td>Boosting + Detectors + CRF (31)</td>
<td></td>
<td>83.8</td>
<td>62.5</td>
</tr>
<tr>
<td>Super Parsing (32)</td>
<td></td>
<td>83.3</td>
<td>51.2</td>
</tr>
</tbody>
</table>

### Graph

- **Red line**: U-Net (4 scaling operations)
- **Green line**: MS-D-Net (200 layers)
- **Validation set**: Dashed line
- **Training set**: Dotted line

![Graph showing performance over epochs](image)
Computational requirements

- All layers are computed in the same way
  - No additional layers/operations (e.g. batchnorm, dropout)
  - Implementation is relatively easy
Goal: Recover 3D interior from 2D X-ray images
Tomographic reconstruction

- Unknown object $x$ is scanned by penetrating waves
- Projections $p$ are measured
- Using $p$, we aim to reconstruct $x$
- Acquisition is modeled by system matrix $W$
  \[ W x = p \]

- Problems: system is ill-posed, underdetermined, and huge
Tomographic reconstruction

- **Direct algorithms**
  - FBP, FDK, ...
  - Analytical inversion

- **Iterative algorithms**
  - SIRT, ART, CGLS, ...
  - Solve linear system

- **Regularized methods**
  - TV-min, wavelets, ...
  - Add regularization term
Problem statement

HQ data ➔ easy ➔ HQ image
Problem statement

HQ data → easy → HQ image → hard

LQ data → hard

Analysis
Problem statement

HQ data
- easy

HQ image
- hard

LQ data
- hard
- easy

LQ image
- VERY hard!

Analysis
Problem statement

HQ data → easy → HQ image → hard → LQ image → easy → LQ data

Analysis
Experiments

- Foam-like simulated objects
- Generate **three** 3D objects (1024³ voxels):
  - 1 for training, 1 for validation, 1 for testing
- Use **high-quality rec** as target for training
- Use **low-quality rec** as network input
Projections & angular range

FBP | MS-D Net | TV-Min

64 projs

45°

(Pelt et al, J. Imaging 2019)
Noise

FBP | MS-D Net | TV-Min

Low abs.

High abs.

(Pelt et al, J. Imaging 2019)
Real-world data

- Fatigue-corrosion data (2160x2560x2560 voxels)
- Use first and last scans as training data
- Shown is an intermediate scan

(Pelt et al, J. Imaging 2019)
Real-world data

- Fatigue-corrosion data (2160x2560x2560 voxels)
- Use first and last scans as training data
- Shown is an intermediate scan

(Pelt et al, J. Imaging 2019)
Problem statement
- Goal: detect cell structures
  - Nuclear envelope, mitochondria, ...

- Use eight manually labeled $512^3$ volumes
  - 6 for training, 1 for validation, 1 for testing

(Pelt & Sethian, PNAS 2019)
Analysis - dental CT

- CBCT scan
- Gold standard
- Snake evolution
- MS-D network

Typical Patient 6

Snake evolution, MS-D network, U-Net, ResNet

Single-object ML

(1) Normal scan  (3) Low-res rec  (4) ROI crop  (6) Train network

(2) Zoomed ROI scan  (5) High-res ROI rec  (7) Apply to low-res rec

(Hendriksen et al. 2019)
Single-object ML

Full low-res 100 \( \mu \text{m} \)  ROI high-res 25 \( \mu \text{m} \)

(Hendriksen et al. 2019)
Single-object ML

- Full low-res 100 μm
- ROI high-res 25 μm
- MS-D Net

(Hendriksen et al. 2019)
Single-object ML

Low-res: 68.26 μm
High-res: 17.07 μm

(Hendriksen et al. 2019)
Conclusions

- **Deep learning** has large potential for scientific images
- Popular existing approaches have **several problems**
- **Mixed-Scale Dense network** for large scientific images
- **Good results** for various imaging problems
- Many interesting and exciting **challenges** remain!
Thank you for your attention!

d.m.pelt@cwi.nl


Training example

- Training with 100 similar images
- Learn parts of **existing algorithms** (Pelt & Batenburg 2013)
- **Learned primal-dual algorithms** (Adler & Oktem 2018)
- Use CNN as a **post-processing operation** (Jin et al 2017)
State-of-the-art: Learned primal-dual

(a) 512 × 512 pixel human phantom
(b) Filtered back-projection (FBP)
PSNR 33.65 dB, SSIM 0.830, 423 ms
(c) Total variation (TV)
PSNR 37.48 dB, SSIM 0.946, 64371 ms

(d) FBP + U-Net denoising
PSNR 41.92 dB, SSIM 0.941, 463 ms
(e) Primal-Dual, linear
PSNR 44.10 dB, SSIM 0.969, 620 ms
(f) Primal-Dual, non-linear
PSNR 43.91 dB, SSIM 0.969, 670 ms
State-of-the-art: FBPConvNet

(Jin et al 2017)
State-of-the-art: FBPCConvNet

(Jin et al 2017)
Results - ssTEM segmentation

- 512x512 pixels, only 30 images for training

2560x2560 tomography images of fiber composite
  - Left: 1024 projections, middle/right: 128 projections

500 slices for training, 100 for validation, from top
  - Result for slice from bottom
## Advantages

<table>
<thead>
<tr>
<th></th>
<th><strong>FBPConvNet</strong></th>
<th><strong>MS-D Net</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>~ 31 million</td>
<td>~ 46 thousand</td>
</tr>
<tr>
<td>Intermediate images</td>
<td>822</td>
<td>100</td>
</tr>
<tr>
<td>Max. theoretical size</td>
<td>~ 1024x1024</td>
<td>~ 4096x4096</td>
</tr>
</tbody>
</table>
## Advantages

<table>
<thead>
<tr>
<th></th>
<th>FBPConvNet</th>
<th>MS-D Net</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td>~ 31 million</td>
<td>~ 46 thousand</td>
</tr>
<tr>
<td><strong>Intermediate images</strong></td>
<td>822</td>
<td>100</td>
</tr>
<tr>
<td><strong>Max. theoretical size</strong></td>
<td>~ 1024x1024</td>
<td>~ 4096x4096</td>
</tr>
</tbody>
</table>

![Graph showing SSIM comparison between FBPConvNet and MS-D Net]

- **SSIM**
  - **Similar**
  - **Improvement**