Machine Learning Approaches for Solving Inverse Imaging Problems

Aggelos K. Katsaggelos

Professor
Joseph Cummings Chair
Northwestern University
Department of EECS
Department of Linguistics
NorthSide University Hospital System
Argonne National Laboratory
Evanston, IL 60208
www.ece.northwestern.edu/~aggk

EUSIPCO, Kos Island, August 29, 2017
Talk Outline

• Inverse Problems
• Analytic Approaches
• Data-Driven Approaches
  – Dictionary Techniques
  – Deep Learning Techniques
• Analytic AND Data-Driven Approaches
• Closing Thoughts
• Compare and contrast analytical and learning approaches in solving inverse problems in imaging
• Review some of the recent advances in solving inverse imaging problems using (deep) learning approaches
• Address their point of intersection
Questions to be Addressed

• Relative advantages of analytical and learning approaches for solving inverse problems, or
• for a specific problem at hand which of the approaches in our available toolbox should one use?
• Under what circumstances would one expect learning approaches to provide more accurate solutions (computational considerations aside) than analytical approaches?
• or would one expect to see the same gains in solving inverse problems with neural networks as we have seen in solving certain classification problems (equaling or even surpassing human performance)?

Known Problem Type

\( f, g \)  
\( f, T \)  
\( T, g \)  
\( g \)  
\( g, T \) 

to partially 

system identification  
system implementation  
recovery -- an inverse problem  
blind recovery  
semi-blind recovery
Forms of the Recovery Problem

- Single vs. multiple observations
- Noise Smoothing
- Restoration/Deconvolution (1D, 2D, 3D)
  - Multi-spectral, multi-channel
- Removal of Compression Artifacts
- Super-Resolution (Pansharpening, Demosaicking)
- Inpainting, Concealment
- Reconstruction
- Dual Exposure Imaging
- Compressive Sensing
- Motion Estimation
- Light-field Reconstruction
Degraded Images
Restored Images
Degraded Images
Restored Images

Deconvolved images using log prior

Software available at http://www.dbabacan.info/BDGSP.php

Degraded Images

Software available at https://sites.google.com/site/fbdhsgp/
Analytical vs Learning Approaches

- **Pros: Analytical**
  - Incorporation of domain knowledge

- **Pros: Learning**
  - Fast Inference

- **Pros: Analytical**
  - Typically require one observation

- **Cons: Learning**
  - Hard to incorporate domain knowledge

- **Cons: Analytical**
  - Typically Slow Inference

- **Cons: Learning**
  - Require large amounts of data
Image Super Resolution

Image Super Resolution

4 images
Image Super Resolution

$$y_k = A \ H_k \ C(s_k) \ x + n_k = B_k(s_k)x + n_k$$

LR observation

PSF

Downsampling

Motion Warping

HR image

Noise

Bayesian Model

• Modeling the uncertainties in the registration parameters

\[ p(s_k) = \mathcal{N}(s_k | \bar{s}_k^p, \Lambda_k^p) \]

• Joint Distribution

\[
p(y, x, \{s_k\}, \alpha_{im}, \{\beta_k\}) = p(y | x, \{s_k\}, \{\beta_k\}) p(x | \alpha_{im}) p(s_k) p(\alpha_{im}) \prod_{k=1}^{L} p(\beta_k)
\]

Experiments

- Downsampling 2x
- Translation Rotation
- 3x3 Uniform Blur Gaussian Noise
- Downsampling 2x

Bicubic
Zomet

RSR
ALG1
ALG2

Exact Knowledge of Motion
Experiments

- Motion parameters corrupted by noise
Lessons Learned

• Prior knowledge critical
  – Means to incorporate it into the problem (via modeling)

• Spatio-temporal adaptivity critical

• Automatic estimation of unknown parameters based on the data very important
Dictionary Techniques

• Fixed vs Designed Dictionaries
• Complete or Overcomplete
• All prior knowledge incorporated by the dictionary
• Advances in dictionaries are coupled with advances in solving sparse problems
• Sparse representations are “extensions” of VQ
• “Related” to NN-based Techniques
A Learning Restoration Approach

A Learning Restoration Approach
Training: $$\min_{D,\{\alpha_i\}} \sum_{i=1}^{N} \|x_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1$$

Testing: $$\min_{\alpha_i} \|x'_i - D\alpha_i\|_2^2 + \lambda \|\alpha_i\|_1$$

Represent as: $$[0, 0, ..., 0, 0.6, 0, ..., 0, 0.8, 0, ..., 0, 0.4, ...]$$

Represent as: $$[0, 0, ..., 0, 1.3, 0, ..., 0, 0.9, 0, ..., 0, 0.3, ...]$$
Single Frame Super-Resolution

Training:
\[
\min_{D^h, D^l, a_i^{tr}} \sum_{i=1}^{N} \|x_i^{tr} - D^h a_i^{tr}\|_2^2 + \|y_i^{tr} - D^l a_i^{tr}\|_2^2 + \lambda \|a_i^{tr}\|_1
\]

Reconstruction:
\[
\min_{z_i} \|y_i - D^l z_i\|_2^2 + \lambda \|z_i\|_1
\]

\[
x_i = D^h z_i
\]

Batch Multiple-frame Super-Resolution

\[
\min_{\alpha_k, \alpha_{k+i}^MC \atop i = -M, ..., N, i \neq 0} \ ||y_k - D^l \alpha_k||^2_2 + \sum_{i=-M, i \neq 0}^{N} \ ||y_{k+i}^{MC} - D^l \alpha_{k+i}^{MC}||^2_2 + \lambda \left( \|\alpha_k\|_1 + \sum_{i=-M, i \neq 0}^{N} \|\alpha_{k+i}^{MC}\|_1 \right) \\
+ \sum_{i=-M, i \neq 0}^{N} \gamma_i \|D^h \alpha_k - D^h \alpha_{k+i}^{MC}\|^2_2 \\
x_k = D^h \alpha_k
\]

Recursive Multiple-frame Super-Resolution

\[
\begin{align*}
\min_{\alpha_k} & \| F y_k - F D_{c*}^l \alpha_k \|_2^2 + \lambda \| \alpha_k \|_1 + \sum_{i=1}^{N} \gamma_j \| D_{c*}^h \alpha_k - x_{k-j}^{MC} \|_2^2 \\
x_k = D_{c*}^h \alpha_k
\end{align*}
\]
## Experimental Results

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene 2</td>
<td>45.27</td>
<td>46.12</td>
<td>46.85</td>
<td>46.53</td>
<td>46.91</td>
</tr>
<tr>
<td>Scene 8</td>
<td>38.18</td>
<td>39.94</td>
<td>40.04</td>
<td>41.08</td>
<td>40.27</td>
</tr>
<tr>
<td>Scene 18</td>
<td>41.43</td>
<td>43.04</td>
<td>43.18</td>
<td>43.28</td>
<td>43.45</td>
</tr>
<tr>
<td>Scene 25</td>
<td>44.40</td>
<td>46.69</td>
<td>46.69</td>
<td>47.38</td>
<td>47.85</td>
</tr>
<tr>
<td>Scene 33</td>
<td>40.22</td>
<td>42.95</td>
<td>42.70</td>
<td>43.37</td>
<td>43.59</td>
</tr>
<tr>
<td>Scene 45</td>
<td>42.43</td>
<td>43.72</td>
<td>43.62</td>
<td>43.89</td>
<td>44.11</td>
</tr>
<tr>
<td>Scene 48</td>
<td>33.90</td>
<td>36.10</td>
<td>35.78</td>
<td>36.24</td>
<td>36.39</td>
</tr>
<tr>
<td>Average</td>
<td>40.83</td>
<td>42.65</td>
<td>42.69</td>
<td>43.11</td>
<td>43.22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene 2</td>
<td>47.41</td>
<td>46.01</td>
<td>46.49</td>
<td>48.26</td>
<td>48.11</td>
</tr>
<tr>
<td>Scene 8</td>
<td>41.08</td>
<td>40.26</td>
<td>39.94</td>
<td>41.62</td>
<td>41.72</td>
</tr>
<tr>
<td>Scene 18</td>
<td>43.94</td>
<td>43.66</td>
<td>43.19</td>
<td>44.73</td>
<td>44.94</td>
</tr>
<tr>
<td>Scene 25</td>
<td>48.29</td>
<td>46.21</td>
<td>46.08</td>
<td>49.57</td>
<td>49.68</td>
</tr>
<tr>
<td>Scene 33</td>
<td>43.83</td>
<td>43.00</td>
<td>42.51</td>
<td>45.33</td>
<td>45.44</td>
</tr>
<tr>
<td>Scene 45</td>
<td>44.32</td>
<td>43.46</td>
<td>43.70</td>
<td>44.76</td>
<td>44.91</td>
</tr>
<tr>
<td>Scene 48</td>
<td>37.48</td>
<td>35.51</td>
<td>34.10</td>
<td>37.56</td>
<td>37.31</td>
</tr>
<tr>
<td>Average</td>
<td>43.76</td>
<td>42.59</td>
<td>42.29</td>
<td>44.55</td>
<td>44.59</td>
</tr>
</tbody>
</table>
Experimental Results
Computational X-Ray Fluorescence (XRF) Imaging using Dictionary Learning

How a Painting is Structured

Final details include the depiction of jewels in the haloes, the white rose, and the mordanted star on the Madonna’s shoulder.

http://www.artcons.udel.edu/about/kress/historic-painting-reconstructions/giotto-di-bondone
Why Imaging of Art?
Different modalities to interrogate structure

- Ultraviolet (280-400 nm)
- Visible (450-750 nm)
- Infrared (750-2000 nm)
- X-ray (sub 10 nm)
XRF maps let you see the invisible!

Hesiod Manuscript (RGB Image)  XRF Map (Iron)

Beam size – Step size 200x200µm²
Dwell time 10ms/point - Att (6)
XRF Re-colorization of Hidden Layers


XRF Scanner Advantages & Disadvantages

- **Advantages**
  - Deep X-Ray penetration into hidden layers
  - Elemental composition for chemical characterization and provenance analysis
  - High spectral resolution (4096 channels)

- **Disadvantages**
  - Low Signal-to-Noise-Ratio (SNR)
  - Slow raster scan process -> Low spatial resolution

A typical XRF scan takes 2sec/pixel -> 20,000 s (6hr) for a 100x100 XRF map)
How can we make scanning Faster?

Observation: The XRF Spectra span a small subspace in $\mathbb{R}^B$

Approach: Use dictionary methods to model XRF spectra using a linear basis decomposition
Vincent Van Gogh, *The Bedroom* (1889)
Art Institute of Chicago 1926.417
XRF Scanning of Van Gogh’s Bedroom
XRF Super-Resolution + RGB Image Fusion

Captured XRF Cube

Super-resolution + Fusion algorithm

Conventional RGB image

Super-resolution HR XRF image

Vincent Van Gogh, *The Bedroom* (1889)
Art Institute of Chicago 1926.417
RGB Fusion using XRF Dictionaries

Input
XRF Cube
\[ X \in \mathbb{R}^{h \cdot w \times B} \]
HR RGB Image
\[ I \in \mathbb{R}^{H \cdot W \times 3} \]

Output
RGB Dictionary
\[ D^{RGB} \in \mathbb{R}^{3 \times N} \]
XRF Dictionary
\[ D^{xrf} \in \mathbb{R}^{B \times N} \]
High Res XRF Cube
\[ Y \in \mathbb{R}^{W \cdot H \times B} \]

Superresolution + Fusion:
\[
D^{rgb^*}, D^{xrf^*}, A^* = \min_{D^{rgb}, D^{xrf}, A} \| I - D^{rgb} A \|^2_F + \| X - D^{xrf} AS \|^2_F
\]
\[ Y_{SR} = D^{xrf^*} A^* \]
Fusion Results

Area scanned using MA-XRF

38x38cm²
Step size: 0.400 cm

Captured XRF Cube $X$

95x95 Pixels

CrK Peak, Bands 611-657

Fused XRF Cube $Y$

475x475 Pixels
Computational XRF Imaging using Dictionary Learning

- **Input (95x95 pix)**
- **Superresolution (475x475 pix)**
- **RGB Fusion (475x475 pix)**

- **Mercury (Bands 1078-1105)**
- **Arsenic (Bands 1243-1290)**
Computational XRF Imaging using Dictionary Learning

Input (95x95 pix)  Superresolution (475x475 pix)  RGB Fusion (475x475 pix)

Mercury (Bands 1078-1105)

Arsenic (Bands 1243-1290)
Computational XRF Imaging using Dictionary Learning

- **Input (95x95 pix)**
- **Superresolution (475x475 pix)**
- **RGB Fusion (475x475 pix)**

**Cobalt (Bands 840-875)**

- Top left: Image showing cobalt distribution.
- Top middle: Superresolved cobalt image.
- Top right: RGB fusion of cobalt image.

**Copper (Bands 875-892)**

- Bottom left: Image showing copper distribution.
- Bottom middle: Superresolved copper image.
- Bottom right: RGB fusion of copper image.
Computational XRF Imaging using Dictionary Learning

Superresolution (475x475 pix)  RGB Fusion (475x475 pix)

Input (95x95 pix)

Cobalt (Bands 840-875)

Copper (Bands 875-892)
What about hidden layers?

Visible

XRF Cube \( X_v \in \mathbb{R}^{h \cdot w \times B} \)

HR RGB Image \( I \in \mathbb{R}^{H \cdot W \times 3} \)

Output

High Res XRF Cube \( Y_v \in \mathbb{R}^{W \cdot H \times B} \)

Not Visible

XRF Cube \( X_{nv} \in \mathbb{R}^{h \cdot w \times B} \)

Output

High Res XRF Cube \( Y_v \in \mathbb{R}^{W \cdot H \times B} \)

Input

\( I \in \mathbb{R}^{H \cdot W \times 3} \)

High Res XRF Cube \( Y \in \mathbb{R}^{W \cdot H \times B} \)
**RGB Fusion Model for Hidden Layers**

**Inputs**
- $X$: Low resolution input XRF image
- $X_v$: Visible part of $X$
- $X_{nv}$: Non-visible part of $X$
- $I$: Input high resolution RGB image
- $S$: Spatial down-sample matrix

**Outputs**
- $D_{nv}^{xrf}$: XRF dictionary (non-visible)
- $D_v^{xrf}$: XRF dictionary (visible)
- $D_{rgb}^{xrf}$: RGB dictionary
- $A_v$: Coefficient of visible component
- $A_{nv}$: Coefficient of non-visible component

**Superresolution + Fusion + V/NV:**

$$
\begin{align*}
D_{rgb}^*, D_{nv}^{xrf}, D_v^{xrf}, A_v, A_{nv} &= \\
&\min_{D_{rgb}^*, D_v^{xrf}, D_{nv}^{xrf}, A_v, A_{nv}} \\
&\frac{1}{2} \| X_{nv} - D_{nv}^{xrf} A_{nv} S \|^2_F + \| \nabla (D_{nv}^{xrf} A_{nv}) \|^2_F \\
&\frac{1}{2} \| I - D_{rgb}^* A_v \|^2_F + \| X_v - D_v^{xrf} A_v S \|^2_F
\end{align*}
$$

**Reconstruction:**

$$
Y = D_{nv}^{xrf} A_{nv}^* + D_v^{xrf} A_v^*
$$
Extensions

• Semi-coupled dictionaries (sparse representations are linearly related)
• Sparse representations share the same support
• Manifold similarity
• Self-dictionaries
• Multi-Scale dictionaries
Neural Network
Function Approximation

\[ f_m(x) = \tanh(c_m + x^T v_m) \quad \text{for all } m \geq 1, \]

\[ f_m(x) = \tanh \left( c_m^{(1)} + \sum_{m_2=1}^{M_2} \tanh \left( c_{m_2}^{(2)} + x^T v_{m_2}^{(2)} \right) v_{m_2,m}^{(1)} \right). \]

\[ f_m(x) = \max \left( 0, c_m^{(1)} + \sum_{m_2=1}^{M_2} \max \left( 0, c_{m_2}^{(2)} + \sum_{m_3=1}^{M_3} \max \left( 0, c_{m_3}^{(3)} + x^T v_{m_3}^{(3)} \right) v_{m_3,m_2}^{(2)} \right) v_{m_2,m}^{(1)} \right). \]
Function Approximation

Figure 5.4. (top row, from left to right) The first four non constant elements of the polynomial basis. (bottom row, from left to right) The first four non constant elements of the Fourier basis.

Figure 5.5. Unlike a fixed basis, elements of an adjustable basis are free to change form by adjusting internal parameters. Here we show four instances of a single (top row) hyperbolic tangent and (bottom row) hinge basis function, with each instance corresponding to a different setting of its internal parameters.
Figure 5.7. Four instances of a two layer neural network basis function made by composing (top row) hyperbolic tangent and (bottom row) hinge functions. In each instance the internal parameters are set randomly. Note how the two layer basis elements are far more diverse in shape than those of a single layer basis shown in Figure 5.5.
Figure 5.6. From left to right, approximation of a continuous function (shown by the dashed black curve) over [0, 1], using $M = 2$ and $M = 6$ elements of (top row) polynomial, (middle row) Fourier, and (bottom row) single hidden layer neural network bases, respectively. While all three bases could approximate this function as finely as desired by increasing $M$, the neural network basis (with its adjustable internal parameters) approximates the underlying function more closely using the same number of basis elements compared to both fixed bases.
CNNs offer multiple advantages over fully-connected NNs:
  – Smaller number of parameters
  – Translation invariance and locality
“Vanilla” CNN

- The “Vanilla” CNN shown in the previous slide has been used for many inverse imaging applications including denoising, super-resolution, and compressive sensing.

Multiple Frame Super Resolution

PSNR for 4x Upscaling:

<table>
<thead>
<tr>
<th>Architecture</th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 frame</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>31.26</td>
</tr>
<tr>
<td>3 frames</td>
<td>31.77</td>
<td><strong>31.81</strong></td>
<td>31.80</td>
<td>31.31</td>
</tr>
<tr>
<td>5 frames</td>
<td>31.80</td>
<td><strong>31.85</strong></td>
<td>31.82</td>
<td>31.33</td>
</tr>
</tbody>
</table>

Filter Symmetry Enforcement

For Architecture (b):

For Architecture (c):
Adaptive MC

Walk Sequence, 3x Upscaling

| Original | Bayesian-MB: 28.37 dB | VSRnet MC: 29.94 dB | VSRnet AMC: 30.21 dB |

Foreman Sequence, 3x Upscaling

| Original | Bayesian-MB: 33.83 dB | VSRnet MC: 34.79 dB | VSRnet AMC: 35.74 dB |
## Experimental Results

<table>
<thead>
<tr>
<th>MYANMAR</th>
<th>Scale</th>
<th>Image SR Algorithms</th>
<th>Video SR Algorithms</th>
<th>Own</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR</td>
<td>2</td>
<td>34.59</td>
<td>-</td>
<td>37.19</td>
</tr>
<tr>
<td>SSIM</td>
<td>2</td>
<td>0.9458</td>
<td>-</td>
<td>0.9638</td>
</tr>
<tr>
<td>PSNR</td>
<td>3</td>
<td>31.59</td>
<td>32.71</td>
<td>33.48</td>
</tr>
<tr>
<td>SSIM</td>
<td>3</td>
<td>0.8957</td>
<td>0.9127</td>
<td>0.9191</td>
</tr>
<tr>
<td>PSNR</td>
<td>4</td>
<td>29.53</td>
<td>-</td>
<td>30.88</td>
</tr>
<tr>
<td>SSIM</td>
<td>4</td>
<td>0.8526</td>
<td>-</td>
<td>0.8777</td>
</tr>
</tbody>
</table>
Bicubic
VSRnet
## CNNs vs. Dictionaries

### Myanmar Video

<table>
<thead>
<tr>
<th></th>
<th>Bicubic</th>
<th>MDMF-B-VT</th>
<th>MDMF-R-VT</th>
<th>VSRnet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scene 2</td>
<td>45.27</td>
<td><strong>48.26</strong></td>
<td>48.11</td>
<td>47.92</td>
</tr>
<tr>
<td>Scene 8</td>
<td>38.18</td>
<td>41.62</td>
<td>41.72</td>
<td><strong>42.53</strong></td>
</tr>
<tr>
<td>Scene 18</td>
<td>41.43</td>
<td>44.73</td>
<td>44.94</td>
<td><strong>45.00</strong></td>
</tr>
<tr>
<td>Scene 25</td>
<td>44.40</td>
<td>49.57</td>
<td><strong>49.68</strong></td>
<td>48.45</td>
</tr>
<tr>
<td>Scene 33</td>
<td>40.22</td>
<td>45.33</td>
<td><strong>45.44</strong></td>
<td>44.50</td>
</tr>
</tbody>
</table>
Skip connections allow for residual learning between the input and output image.

Results

- Residual learning can lead to a significant improvement over “vanilla” end-to-end mapping.
Improvements over vanilla CNN: Residual blocks

- Residual blocks help train networks with greater depth – increasing the capacity of the neural network.
Improvements over vanilla CNN: Encoder-decoder CNNs

- The first part of the network, the "compressive" part, learns an abstract representation of the input image, which is then used by the "expansive" part of the network to produce an output image.
- This modelling has a very intuitive justification in the probabilistic formulation of the inverse problem, in which we find a set of latent variables which after decoding are able to explain our observations.
Improvements over vanilla CNN: Encoder-decoder CNNs

- Encoder-decoder CNNs have been used in image inpainting, denoising, and optical flow estimation.

High Speed Compressive Video

1. Capture high-speed video using a low frame-rate camera.
2. Optimize utilized sampling patterns.
**CS Video System**

- **Measurement model:**
- **Tested masks:**
  - Shifted-Normalized masks provided superior performance to Shifted-Random.
NU CS Video System

Proposed Encoder-Decoder

Encoder: we learn the **binary** weights of the sensing matrix
Decoder: we learn to reconstruct the video sequence

Quantitative Results

The image shows bar charts comparing the performance of different reconstruction algorithms under varying initial percentages of 1's on the mask. The charts display the average PSNR (纵轴) and SSIM (横轴) for each algorithm (FC4-I, GMM-1, GMM-4, DICT-L1, TV-MIN) under four different initial percentages of 1's on the mask: 20%, 40%, 60%, and 80%. The performance metrics are evaluated against two different initial binary masks: Initial binary mask and Optimized binary mask.
Video Reconstruction Results

DICT-L1, DeepMask-40, PSNR: 30.05

DICT-L1, RandomMask-40, PSNR: 28.42

TV-MIN, DeepMask-40, PSNR: 29.86

TV-MIN, RandomMask-40, PSNR: 28.07
DICT-L1, DM-40, PSNR: 27.88

GMM-4, DM-40, PSNR: 28.79

TV-MIN, DM-40, PSNR: 28.14

FC4-1M, DM-40, PSNR: 30.09
An Inpainting Approach

Full Scan: 580x680=394400 pixels

20% Scan: 580x680x20%=78880 pixels
5X speedup
An Inpainting Approach

Full Scan: 580x680=394400 pixels

20% Scan: 580x680x20%=78880 pixels
5X speedup

Adaptive Sampling Mask
Experimental Results: RGB Image Inpainting

Random Sampling Mask
PSNR: 26.97 dB
Experimental Results: RGB Image Inpainting

Adaptive Sampling Mask
PSNR: 29.04 dB
Proposed Method

\[\min_{NetM,NetE} \sum_{i=1}^{N} \|\bar{x}_i - x_i\|_2^2\]
XRF/RGB Fusion Inpainting

1. Conventional Camera
   - Conventional RGB image

2. XRF Scanner
   - Measured Sparse XRF image

3. NetM
   - Sparse Sampling Mask

4. Proposed Inpainting Algorithm

5. Reconstructed XRF image
MS reconstruction, Channel #13, Random sampling mask
MS reconstruction, Channel #13, Adaptive sampling mask
Ground Truth Channel #13
XRF/RGB Fusion Inpainting
Fusion Result of Random Sampling Channel #13
XRF/RGB Fusion Inpainting
Fusion Result of Adaptive Sampling Channel #13
Ground Truth Channel #13
Because CNNs are able to learn all the relevant statistics of natural images, feature space losses can be used to solve inverse imaging problems.
From MSE to feature-space loss

Feature-space loss for super-resolution

Ground Truth  
This image

<table>
<thead>
<tr>
<th>Method</th>
<th>l_{pixel}</th>
<th>l_{feat}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bicubic</td>
<td>31.78</td>
<td>0.8577</td>
</tr>
<tr>
<td>Ours (l_{pixel})</td>
<td>31.47</td>
<td>0.8573</td>
</tr>
<tr>
<td>SRCNN [11]</td>
<td>32.99</td>
<td>0.8784</td>
</tr>
<tr>
<td>Ours (l_{feat})</td>
<td>29.24</td>
<td>0.7841</td>
</tr>
</tbody>
</table>

End-to-end mapping in learned feature spaces

Training phase 1: train autoencoders to learn new representations

Training phase 2: train a neural network to map one representation to another

Testing (inference) phase

Generative Adversarial Networks (GANs) are composed of two models:

- The **generative model** is the model generating a sample $G(z)$
- Given $G(z)$, the **discriminative model** learns to discriminate samples that come from the generator with samples from the training data. It does so by outputting a probability $D(G(z))$.

---

Conditional Generative Adversarial Networks (cGANs)

- Generative Adversarial Networks (GANs) can learn distributions of the training data.

\[ l_{GAN} = \mathbb{E}_{x \sim p_{data}(x)} \left[ \log(D(x)) \right] + \mathbb{E}_{y \sim p_{data}(y)} \left[ \log(1 - (D(G(y)))) \right] \]
Solution Pushed to Natural Image Manifold

"pixel-wise average of possible solutions"

Natural Image Manifold
MSE-based Solution
GAN-based Solution
cGANs for image super-resolution

original
bicubic (21.59dB/0.6423)
SRResNet (23.44dB/0.7777)
SRGAN (20.34dB/0.6562)

Using NNs as denoiser in variable splitting-based optimization

ADMM recovery formulation

\[
\min_{x,z} \frac{1}{2} \| y - Az \|^2 + \lambda \phi(x)
\]
\[\text{s.t } x = z\]

Augmented Lagrangian

\[
\mathcal{L}(x, z, u) = \frac{1}{2} \| y - Az \|^2 + \lambda \phi(x) + \frac{\rho}{2} \| x - z - u \|^2
\]

Iterative procedure

\[
x^{k+1} = \arg \min_x \frac{\rho}{2} \| x - z^k + u^k \|^2 + \lambda \phi(x)
\]
\[
z^{k+1} = \arg \min_x \frac{1}{2} \| y - Az^k \|^2 + \frac{\rho}{2} \| x^{k+1} - z + u^k \|^2
\]
\[
u^{k+1} = u^k + x^{k+1} - z^{k+1}
\]

Update to be replaced by a DNN
Learning a single image prior

Learning a proximal operator with a deep projection model

Variable splitting and NNs

- In variable splitting method, we separate our problem into a denoising problem and a fidelity problem.

- The solution to the denoising problem can be obtained by a CNN. The regularization term in the denoising sub-problem is learned from the dataset by the CNN.

\[
x_{k+1} = \arg \min_x \| y - Hx \|^2 + \mu \| x - z_k \|^2
\]

Fidelity sub-problem

\[
z_{k+1} = \arg \min_z \frac{\mu}{2} \| z - x_{k+1} \|^2 + \lambda \Phi(z)
\]

Denoising sub-problem

Variable splitting and NNs

- The variable splitting with NN approach has the advantage that it can be applied to multiple inverse imaging problems, including image denoising, image deblurring, and image super-resolution.

Conditional Generative Adversarial Networks (cGANs)

Results in [1] indicate that when the process to obtain the LR image from the HR one is known, this knowledge leads to better HR reconstructions. In [2]

\[ y = \Delta x \]

The generator has the form (notice that it takes into account the observation process)

\[ g_\phi(y) = (I - A^+A)f_\phi(y) + A^+y \]

The generator is updated by minimizing

\[ l_{GAN}(G, D) = \mathbb{E}_{y \sim p_{data}(y)} \log \frac{1 - D(g_\phi(y))}{D(y)} \]


Deep Unfolding

• Model-based methods: problem domain knowledge can be built into the constraints of the model, typically at the expense of difficulties during inference.
• Deterministic deep neural networks: inference is straightforward but their architectures are generic and it is unclear how to incorporate knowledge.
• “Solution”: start with a model-based approach and an associated inference algorithm and unfold the inference iterations as layers in a deep network.
• In addition, instead of optimizing the original model, the model parameters are untied across layers, so as to create a more powerful network.

The Lasso Problem and ISTA/FISTA

\[ z^* = \arg\min_z ||x - Dz||_2^2 + \lambda ||z||_1 \]

\[ z(k + 1) = h_\theta(W \cdot x + S \cdot z(k)) \]

\[ W = \frac{1}{L} D^T \]

\[ S = I - \frac{1}{L} D^T D \]

The shrinkage function \( h_\theta \) is defined as
\[ [h_\theta(Y)]_i = \text{sign}(Y_i)(|Y_i| - \theta_i)^+ \]

\( L \) is the upper bound on the largest eigenvalue of \( D^T D \)
Learned ISTA (LISTA)

\[ z = L(x, W, S, \theta) \]

Learned ISTA

\[ z = L(x, W, S, \theta) \]

Unlike ISTA, where \( W, S, \theta \) are determined by \( D, \lambda \), Learned ISTA learn the network parameter \( W, S, \theta \) by minimizing

\[
\min_{W,S,\theta} \sum_{j=1}^{N} \| L(x_j, W, S, \theta) - z_j^* \|^2_2
\]

\[
z^* = \arg\min_z \| x - Dz \|^2_2 + \lambda \| z \|_1
\]
Unfolding for MRI compressive sensing

- Here we base the graph structure on the ADMM algorithm, and learn the parameters as the layers of the neural network.
- The neural network parameters to be learned include a nonlinear transformation of the CS measurements, the shrinkage function, the regularization function, in addition to the various hyper-parameters of the ADMM algorithm.

Final Thoughts

• Learning in general and via DNNs in particular, although not precisely understood, provides fair competition to modeling techniques, capitalizing on the power of computers (e.g., GPUs) and the abundance of data

• The impressive achievements in recognition tasks (face, speech) do not transfer to recovery problems (yet)

• Advantageous in more “complicated” recovery cases where modeling is at a disadvantage
Final Thoughts

• In spite of the success, by and large DL for image recovery does not usually capture model uncertainty. Most of the current DL approaches can be thought of as ML or MAP methods.

• On the other hand, analytical techniques have greatly benefited from capturing model uncertainty and using approximate inference methods like VB and EP.

• The introduction of GANs (and VAEs) and the description of fundamental DL techniques such as dropout as VB approximation (*) will likely lead to a better modeling of uncertainty with DL techniques for imaging problems.

Contributors

• Prof. Rafael Molina, U Granada
• Prof. Ollie Cossairt, Northwestern
• Prof. Marc Walton, Northwestern
• Dr. P. Ruiz, Northwestern
• Dr. Derin Babacan, Google
• Dr. Zhao Fu Chen, Google
• Dr. Michael Iliadis, Sony
• Dr. Leonidas Spinoulas, ISI/USC
• Dr. Armin Kappeler, Yahoo
• Tim Qai, EyeVerify
• Andrew Yoo, Northwestern