Single Image Super Resolution

Wenchao Zheng
5/6/2016
MCL
University of Southern California
Outline

Introduction
• Definition of Super Resolution
• Multi-image SR
• Single-Image SR

Literature
• Example based
  • SRCNN
  • VDSR
  • Perceptual SR

Current Work
• Match filter
Outline

**Introduction**
- Definition of Super Resolution
- Multi-image SR
- Single-Image SR

**Literature**
- Example based
  - SRCNN
  - VDSR
  - Perceptual SR

**Current Work**
- Match filter
Definition:
Generating high resolution (HR) image with more details using one or more low resolution (LR) images.
More examples
More examples

Low-res image  
Super resolved image
Difference with Sharpening
Multi-image vs. Single-image

Multi-image

Source: [Park et al. SPM 2003]

Single-image

Source: [Freeman et al. CG&A 2002]
Multi-image SR

- Several images of the same scenery.
- Each image has different information of the same scenery.
Multi-image SR

- Reconstruct HR image through multiple LR images
Single Image SR

Learning to map from low-res to high-res patches

- Nearest neighbor [Freeman et al. CG&A 02]
- Neighborhood embedding [Chang et al. CVPR 04]
- Sparse representation [Yang et al. TIP 10]
- Kernel ridge regression [Kim and Kwon PAMI 10]
- Locally-linear regression [Yang and Yang ICCV 13] [Timofte et al. ACCV 14]
- Convolutional neural network [Dong et al. ECCV 14]
- Random forest [Schulter et al. CVPR 15]
Single Image SR

Low Resolution  Super Resolution  Original
Outline

Introduction
• Definition of Super Resolution
• Multi-image SR
• Single-Image SR

Literature
• Example based
• SRCNN
• VDSR
• Perceptual SR

Current Work
• Match filter
Example Based SR

Correspondences between low and high resolution image patches are learned from a database of low and high resolution image.
Example Based SR
**Example Based SR**

- **Super-Resolution from Transformed Self-Exemplars**

Example Based SR

LR input image

Matching

LR patch

HR patch

Translation

Ground truth

LR/HR patch

Perspective
A review on: *Image Super-Resolution using deep CNN*

- First super resolution topic using CNN
- Achieved significant improvement

Training: Network overview

Low-resolution image (input)

Patch extraction and representation

Non-linear mapping

Reconstruction

backpropagation
• **Patch extraction & representation**
  - 64 filters of size $C \times f_1 \times f_1$ (1x9x9) \[64 \times 1 \times 9 \times 9 = 5184\]

• **Non-linear mapping**
  - 32 filters of size $n_1 \times f_2 \times f_2$ (64x1x1) \[32 \times 64 \times 1 \times 1 = 2048\]

• **Reconstruction**
  - 1 filters of size $n_2 \times f_3 \times f_3$ (32x5x5) \[1 \times 32 \times 5 \times 5 = 800\]
Training: preprocessing

• Training dataset
  - 91 images
  - only use Y channel

• Input patch extraction
  - extract patches from bicubic interpolated LR images
  - input patches are of size 33x33
  - normalize pixel value to 0~1

• Target patch extraction
  - extract patches from original HR images
  - target patches are of size 21x21
  - normalize pixel value to 0~1
Training: make LR match HR

Euclidian distance

\[ L(\Theta) = \frac{1}{n} \sum_{i=1}^{n} \| F(Y_i; \Theta) - X_i \|^2, \]
Training: filter visualization

$g$ and $h$ are like Laplacian/Gaussian filters
$a$ – $e$ are like edge detectors at different directions
$f$ is like a texture extractor
Training: feature maps

Input

Output

Feature maps of the first layer

Feature maps of the second layer
Training: number of layers

(a) 9-1-5 vs. 9-1-1-5
(b) 9-3-5 vs. 9-3-1-5
(c) 9-5-5 vs. 9-5-1-5
### Training: number of filter

<table>
<thead>
<tr>
<th>$n_1 = 128$</th>
<th>$n_1 = 64$</th>
<th>$n_1 = 32$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_2 = 64$</td>
<td>$n_2 = 32$</td>
<td>$n_2 = 16$</td>
</tr>
<tr>
<td>PSNR Time (sec)</td>
<td>PSNR Time (sec)</td>
<td>PSNR Time (sec)</td>
</tr>
<tr>
<td>32.60 0.60</td>
<td>32.52 0.18</td>
<td>32.26 0.05</td>
</tr>
</tbody>
</table>
Test: forward propagation

Using full bicubic interpolated image as input, not patch
Performance comparison

![Graph showing performance comparison with different methods.
- SRCNN: 33 dB
- A+: 32.59 dB
- KK: 32.28 dB
- ANR: 31.92 dB
- NE+LLE: 31.84 dB
- SC: 31.42 dB
- Bicubic: 30.39 dB

Average test PSNR (dB) vs. Number of backprops (10^8).]
Performance comparison

Table 1: performance on BSD200

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PSNR</strong></td>
<td>2</td>
<td>28.38</td>
<td>-</td>
<td>29.67</td>
<td>30.02</td>
<td>29.72</td>
<td>30.14</td>
<td>30.29</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>25.94</td>
<td>26.54</td>
<td>26.67</td>
<td>26.89</td>
<td>26.72</td>
<td>27.05</td>
<td>27.18</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>24.65</td>
<td>-</td>
<td>25.21</td>
<td>25.38</td>
<td>25.25</td>
<td>25.51</td>
<td>25.60</td>
</tr>
<tr>
<td><strong>SSIM</strong></td>
<td>2</td>
<td>0.8524</td>
<td>-</td>
<td>0.8886</td>
<td>0.8935</td>
<td>0.8900</td>
<td>0.8966</td>
<td>0.8977</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.7469</td>
<td>0.7729</td>
<td>0.7823</td>
<td>0.7881</td>
<td>0.7843</td>
<td>0.7945</td>
<td>0.7971</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>0.6727</td>
<td>-</td>
<td>0.7037</td>
<td>0.7093</td>
<td>0.7060</td>
<td>0.7171</td>
<td>0.7184</td>
</tr>
</tbody>
</table>
PSNR: Bicubic 32.39 dB and SRCNN 34.71 dB
Improvement: 2.32 dB
SRCNN’s Strength

PSNR: Bicubic 32.58 dB and SRCNN 35.48 dB
Improvement: 2.90 dB
SRCNN’s Strength

PSNR: Bicubic 24.04 dB and SRCNN 27.95 dB
Improvement: 3.94 dB
SRCNN’s weakness

PSNR: Bicubic 33.91 dB and SRCNN 35.25 dB
Improvement: 1.34 dB
SRCNN’s weakness

PSNR: Bicubic 28.05 dB and SRCNN 29.64 dB
Improvement: 1.59 dB
Modify SRCNN’s training data
SRCNN trained on hair dataset

Author-given training dataset

Specific Hair training dataset

PSNR 30.60 dB  SSIM 0.9572

PSNR 30.90 dB  SSIM 0.9599
SRCNN trained on hair dataset

Author-given training dataset

Specific Hair training dataset

32.38 dB
0.8511

32.45 dB
0.8531
 SRCNN trained on hair dataset

Author-given training dataset

Specific Hair training dataset

<table>
<thead>
<tr>
<th></th>
<th>30.19 dB</th>
<th>0.8256</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRCNN trained on hair dataset</td>
<td>30.21dB</td>
<td>0.8240</td>
</tr>
</tbody>
</table>
A review on: *Accurate Image Super-Resolution Using Very Deep Convolutional Networks*

- Second super resolution topic using CNN with deeper network
- Less training iteration
- Less training time
- Achieved huge improvement

Training: network overview

- Residue learning
- 20 layers
- Large learning rate 0.1
Some questions

• Why the deeper the better?
• Why learning error map is better?
• Why large learning rate? And how to achieve convergence?
• How to choose filter size?
Why the deeper the better?

- SRCNN only has three layers
  - They stopped the training procedure before networks converged
  - Learning rate 10\(^{-5}\) is too small for a network to converge within a week on a common GPU.

- CNN exploits spatially local correlation.

- Stacking many layers leads to filters that become increasingly global.

Why the deeper the better?
Why learning error map is better?

- Advantages
  - Converges faster
  - Superior results

(a) Initial learning rate 0.1
(b) Initial learning rate 0.01
(c) Initial learning rate 0.001
Why large learning rate?

• large learning rate can boost training

• Initially in SRCNN, the learning rate is 0.0001, and the total iteration is 15,000,000.

• How to assure convergence?
Large learning rate issue

- Gradient Explosion

stochastic gradient descent equation:

\[ V_{t+1} = \mu V_t - \alpha \nabla L(W_t) \]
\[ W_{t+1} = W_t + V_{t+1} \]
Gradient Clipping

- Clip weight updates value to a predefined small range $[-\theta, \theta]$.

- Since gradient is multiplied by learning rate to update weights.
  
  $$V_{t+1} = \mu V_t - \alpha \nabla L(W_t)$$
  
  $$W_{t+1} = W_t + V_{t+1}$$

- Gradients are actually clipped to
  
  $$[-\frac{\theta}{\gamma}, \frac{\theta}{\gamma}]$$

- where $\gamma$ is the current learning rate.
Performance of VDSR

- Network trained much faster than SRCNN (4 days -> 3 hours)
- Superior performance than bicubic and SRCNN
• Extremely well in sharp edge refinement
(Cont’d)

• +0.5 dB

Table 1: Comparison of PSNR using different methods

<table>
<thead>
<tr>
<th>Set5</th>
<th>Bicubic (dB)</th>
<th>SRCNN (dB)</th>
<th>VDSR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bird</td>
<td>32.47</td>
<td>34.91</td>
<td>35.70</td>
</tr>
<tr>
<td>Head</td>
<td>32.92</td>
<td>33.55</td>
<td>33.68</td>
</tr>
<tr>
<td>Woman</td>
<td>28.60</td>
<td>30.92</td>
<td>31.47</td>
</tr>
<tr>
<td>Baby</td>
<td>33.95</td>
<td>35.01</td>
<td>35.18</td>
</tr>
<tr>
<td>Butterfly</td>
<td>24.05</td>
<td>27.58</td>
<td>28.74</td>
</tr>
<tr>
<td>Average</td>
<td>30.39</td>
<td>32.39</td>
<td>32.95</td>
</tr>
</tbody>
</table>
A review on: *Perceptual Losses for Real-Time Style Transfer and Super-Resolution*

- PSNR doesn’t guarantee perceptually comfortable SR
- Teach the network to redraw!

network overview
Modified loss function

Feature Reconstruction Loss

\[ \ell_{\text{feat}}^j (\hat{y}, y) = \frac{1}{C_j H_j W_j} \| \phi_j (\hat{y}) - \phi_j (y) \|_2^2 \]

Style Reconstruction Loss

\[ G_j^\phi (x) = \psi \psi^T / C_j H_j W_j \]

\[ \ell_{\text{style}}^j (\hat{y}, y) = \| G_j^\phi (\hat{y}) - G_j^\phi (y) \|_F^2 \]

Final Loss function

\[ \hat{y} = \arg \min_y \lambda_c \ell_{\text{feat}}^j (y, y_c) + \lambda_s \ell_{\text{style}}^j (y, y_s) + \lambda_{TV} \ell_{TV} (y) \]
Style
*The Starry Night*,
Vincent van Gogh,
1889

Style
*The Muse*,
Pablo Picasso,
1935
Style
Composition VII, Wassily Kandinsky, 1913

Style
The Great Wave off Kanagawa, Hokusai, 1829-1832
## Experiment results

<table>
<thead>
<tr>
<th></th>
<th>Ground Truth</th>
<th>Bicubic</th>
<th>Ours ($\ell_{\text{pixel}}$)</th>
<th>SRCNN [11]</th>
<th>Ours ($\ell_{\text{feat}}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>This image</td>
<td>31.78 / 0.8577</td>
<td>31.47 / 0.8573</td>
<td></td>
<td>32.99 / 0.8784</td>
<td>29.24 / 0.7841</td>
</tr>
<tr>
<td>Set5 mean</td>
<td>28.43 / 0.8114</td>
<td>28.40 / 0.8205</td>
<td></td>
<td>30.48 / 0.8628</td>
<td>27.09 / 0.7680</td>
</tr>
</tbody>
</table>
## Experiment results

<table>
<thead>
<tr>
<th></th>
<th>Ground Truth</th>
<th>Bicubic</th>
<th>Ours ((\ell_{\text{pixel}}))</th>
<th>SRCNN [11]</th>
<th>Ours ((\ell_{\text{feat}}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>This Image</td>
<td>21.69 / 0.5840</td>
<td>21.66 / 0.5881</td>
<td>22.53 / 0.6524</td>
<td>21.04 / 0.6116</td>
<td></td>
</tr>
<tr>
<td>Set14 mean</td>
<td>25.99 / 0.7301</td>
<td>25.75 / 0.6994</td>
<td>27.49 / 0.7503</td>
<td>24.99 / 0.6731</td>
<td></td>
</tr>
<tr>
<td>BSD100 mean</td>
<td>25.96 / 0.682</td>
<td>25.91 / 0.6680</td>
<td>26.90 / 0.7101</td>
<td>24.95 / 63.17</td>
<td></td>
</tr>
</tbody>
</table>
Outline

Introduction
- Definition of Super Resolution
- Multi-image SR
- Single-Image SR

Literature
- Example based
  - SRCNN
  - VDSR
- Perceptual SR

Current Work
- Match filter
Matched Filter Based SR

- Input patch
- Match filters
- Conv results
- ReLu
- SVR
- Final prediction
Matched filters

Bicubic patch  Original patch  prediction
Matched filters

- Cluster patches based on prediction and patch similarity