Pixel Recursive Super Resolution
by Google Brain
Bilkent University
CS 559
Diaa Badawi
Outline

• Introduction
• SR Algorithms types
• Related Work
• Pixel Recursive Super Resolution
• Results
• Conclusion
• References
Introduction: What is SR?

- SR is the process of obtaining one or more high-resolution images from one or more low-resolution images.
- The magnification factor is >2.
Introduction: Why is it hard?

• SR is an ill-posed inverse problem.
• LR images contain less information than HR images.
• Therefore, we need to infer the finer details.
• The problem is underspecified (too many possible solutions).
Introduction: Why is it hard?

• Only very few are plausible.
• It becomes much harder with larger target magnification factors.
• From Probabilistic Perspective, model should account for multi-modality.
Introduction: Why is it hard?

- What is a plausible solution?
  - Plausible solution should look natural to human eye.
  - It needs to account for complex structures of the image.
- How do we know whether a solution is plausible?
  - Human perceptual assessment.
  - pSNR and MSE are not sophisticated enough; they may be misleading.
  - Still an open research problem (so is SR problem).
Introduction: Why is it hard?

Down-sampled image

True image

??
Introduction: Why is it important?

• In reality, we want to obtain HR presentation from existing or forthcoming LR data.

• Upgrading sensory hardware?
  • Might be expensive.
  • Might not be currently possible.

• E.g. Decreasing pixel cell size beyond a threshold decreases the amount of light reaching the cell -> decreases SNR.
Introduction: Application Areas

• Satellite and aerial imaging.
• Medical Imaging.
• Text images improvement.
• Compressed image and video enhancement.
• Many more ...
Introduction: Application Areas
Introduction: Application Areas
SR Algorithm Types

• SR problem can be classified into:
  • Spatial Domain
  • Frequency Domain (less popular)

• Also:
  • Single Image Super Resolution (SISR).
  • Multiple Image Super Resolution (MISR).
SR and Deep Learning

• From ML viewpoint, SR is an unsupervised task.
• Encoding the complete set of dependencies between pixels.
• Modelling the multimodality of the problem.
Previous Work

- There is numerous research into SR problem dating as far back as 1970s.
- Examples are:
  - “Image super-resolution as sparse representation of raw image patches” by J Yang et. al.
  - “Learning a Deep Convolutional Network for Image Super-Resolution” by Dong et. al.
SRCNN

- One of the first deep-learning SR frameworks.
- Image is first up-scaled using bi-cubical interpolation.
- The up-scale image is then fed to a convolutional NN.
SRCNN

Direct Relation with Sparse-Coding method
Related Work

• ‘Pixel Recursive super Resolution’ is built upon two Deep learning Frameworks:
  • **PixelCNN/PixelRNN** in “Pixel recurrent neural networks” by Oord *et al*.
  • **SRResNet** in “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network” by Ledig *et al*.
PixelCNN/PixelRNN

- Large-scale modelling of natural images.
- Modelling complete set of dependencies in the image.
- Treats Pixels as discrete random variables -> multinomial distribution.
- Unsupervised.
- Generative NNs.

*Figure 1. Image completions sampled from a PixelRNN.*
PixelCNN/PixelRNN

- Raster Scanning.
- \( p(x_i | x_{<i}) = p(x_{i,R} | x_{<i})p(x_{i,G} | x_{<i}, x_{i,R})p(x_{i,B} | x_{<i}, x_{i,R}, x_{i,G}) \)
- It uses masked convolution:
  - conditions color channels properly.
  - restricts pixel prediction from top and left pixels.
PixelRNN

- Recurrent Model (LSTM).
- Two variants:
  - Row LSTM
  - Diagonal LSTM
- First layer: Uses mask A (spatial convolution)
- Residual Connections.
PixelCNN

- Much Faster than PixelRNN
- No recurrent connections.
- First layer: Uses mask A (spatial convolution).
- Next layers: Uses mask B (spatial convolution) + Residual connections.
- No pooling.
- Enhancement is “Gated PixelCNN”.
PixelCNN/PixelRNN

PixelCNN  Row LSTM  Diagonal BiLSTM

*Figure 4.* Visualization of the input-to-state and state-to-state mappings for the three proposed architectures.
SRResNet

- SRResNet is a generative adversarial NN which employs deep residual network with skip connections.
- It takes a LR image and produces a HR image.
- Batch Normalization
- LeakyReLU
- LR is magnified using Transposed Convolution.
SRResNet

**Generator Network**

- Input
- Conv
- PReLU
- k9n64s1

- B residual blocks
  - k3n64s1
  - k3n64s1
  - Elementwise Sum

- Conv
- BN
- PReLU
- k3n64s1

- Conv
- BN
- PReLU
- PixelShuffler A2
- PReLU
- k3n256s1

- Conv
- PReLU
- k9n3s1

**Skip Connection**

**Discriminator Network**

- Input
- Conv
- Leaky ReLU
- k3n64s1

- Conv
- BN
- Leaky ReLU
- k3n64s2

- k3n128s2
- k3n256s2
- k3n512s2

- Dense [1024]
- Leaky ReLU
- Dense [1]

- Output
Pixel Recursive Super Resolution: General Idea

- SISR deep-learning-based framework.
- Developed by Dahl, Norouzi and Shlens from Google Brain.
- General idea:
  - Using PixelCNN network for encoding prior information in HR space.
  - Using a network similar to SRResNet as a conditioning network for LR image.
  - Train both networks jointly.
- Obtain LR from HR data (simply by down-sampling).
Pixel Recursive Super Resolution: General Idea

• Conditional dependence between LR pixels and HR pixels fails to provide powerful encoding.

\[ \log p(y|x) = \sum_{i=1}^{M} \log p(y_i|x) < - \text{cannot encode HR inter-pixel dependencies} \]

• In this model, conditioning dependency is established between:
  • LR pixels and HR pixels
  • Previously predicted HR pixels and HR pixels to be predicted.

• This is modelled as:

\[ \log p(y|x) = \sum_{i=1}^{M} \log p(y_i|x, y_{<i}) \]

• Using PixelCNN, this model can capture multimodality.
Pixel Recursive Super Resolution: Multimodality Problem

How the dataset was created

Samples from trained model

$L_2$ regression

cross-entropy

PixelCNN
Pixel Recursive Super Resolution: Conditioning Problem

• According to the paper, PixelCNN alone tended to ignore the conditioning of the LR image.
• Correlation between SR pixels is stronger than that between SR and LR pixels.
• In order to force the conditioning of LR pixels, SRResNet was incorporated in the scheme.
Pixel Recursive Super Resolution: diagram
Fusion model

• Each network predicts a vector of \( K \) possible values. E.g. \( K = 256 \).
• The probabilistic model for the \( i \)th pixel is the late fusion of the logits of the two models.
• Late fusion -> summation of predictions at the end.
• Soft-max operator is then applied.
• \( p(y_i|x, y_{<i}) = \text{softmax} (A_i(x) + B_i(y_{<i})) \)
• \( A \) is the output of the conditioning network (SRResNet-like).
• \( B \) is the output of the prior network(PixelCNN).
Optimization model

• $o_1 = \sum \sum_{i=1}^{M} 1 \ [y_i^*]^T (A_i(x) + B_i(y_{<i}^*)) - lse (A_i(x) + B_i(y_{<i}^*))$

• To be maximized using stochastic gradient ascent.

• $1$ is a one-hot vector.

• This model ignores the conditioning network.

• $o_2 = \sum \sum_{i=1}^{M} 1 \ [y_i^*]^T (2A_i(x) + B_i(y_{<i}^*)) - lse (A_i(x)$
Optimization model

• Instead of using softmax, the authors use tempered softmax.

\[ p_\tau = \frac{\frac{1}{p^\tau}}{\frac{1}{|p^\tau|}_1} \]

• This gives more freedom to generate different pixel values.
Implementation Details

<table>
<thead>
<tr>
<th>Operation</th>
<th>Kernel</th>
<th>Strides</th>
<th>Feature maps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conditional network – 8 x 8 x 3 input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$B \times$ ResNet block</td>
<td>3 x 3</td>
<td>1</td>
<td>32</td>
</tr>
<tr>
<td>Transposed Convolution</td>
<td>3 x 3</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>$B \times$ ResNet block</td>
<td>3 x 3</td>
<td>1</td>
<td>32</td>
</tr>
<tr>
<td>Transposed Convolution</td>
<td>3 x 3</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>$B \times$ ResNet block</td>
<td>3 x 3</td>
<td>1</td>
<td>32</td>
</tr>
<tr>
<td>Convolution</td>
<td>1 x 1</td>
<td>1</td>
<td>3 x 256</td>
</tr>
<tr>
<td>PixelCNN network – 32 x 32 x 3 input</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Masked Convolution</td>
<td>7 x 7</td>
<td>1</td>
<td>64</td>
</tr>
<tr>
<td>20 x Gated Convolution Blocks</td>
<td>5 x 5</td>
<td>1</td>
<td>64</td>
</tr>
<tr>
<td>Masked Convolution</td>
<td>1 x 1</td>
<td>1</td>
<td>1024</td>
</tr>
<tr>
<td>Masked Convolution</td>
<td>1 x 1</td>
<td>1</td>
<td>3 x 256</td>
</tr>
</tbody>
</table>

| Optimizer                        | RMSProp (decay=0.95, momentum=0.9, epsilon=1e-8) |
| Batch size                       | 32                                               |
| Iterations                       | 2,000,000 for Bedrooms, 200,000 for faces.       |
| Learning Rate                    | 0.0004 and divide by 2 every 500000 steps.        |
| Weight, bias initialization      | truncated normal (stddev=0.1), Constant(0)        |
Experimental Results (LSUN Bedrooms)
Experimental Results (CelebA faces)

<table>
<thead>
<tr>
<th>Input</th>
<th>Bicubic</th>
<th>ResNet $L_2$</th>
<th>$\tau = 1.0$</th>
<th>$\tau = 0.9$</th>
<th>$\tau = 0.8$</th>
<th>Truth</th>
<th>Nearest N.</th>
<th>GAN [11]</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Input Image" /></td>
<td><img src="image2.png" alt="Bicubic Image" /></td>
<td><img src="image3.png" alt="ResNet L2 Image" /></td>
<td><img src="image4.png" alt="tau=1.0 Image" /></td>
<td><img src="image5.png" alt="tau=0.9 Image" /></td>
<td><img src="image6.png" alt="tau=0.8 Image" /></td>
<td><img src="image7.png" alt="Truth Image" /></td>
<td><img src="image8.png" alt="Nearest Neighbors Image" /></td>
<td><img src="image9.png" alt="GAN Image" /></td>
</tr>
</tbody>
</table>
Experimental Result

- Human perceptual evaluation.
- *"Which image, would you guess, is from a camera?"*
- 283 volunteers.
- 1 sec. to decide.
Experimental Result

In Theory, a perfect generator would totally deceive humans such that the chances of synthetic SR image to win against the true image is 50% (random choice)
Conclusion: Strengths (IMO)

- Very good performance on the data sets.
- Ability to capture multi-modality in the data.
- Trained end-to-end (no need to hand-craft the model).
- Success in SR over a large magnification factor (4).
Conclusion: Weaknesses (IMO)

• It is not clear whether the lack of a discriminator network would result in a better correlation between LR and HR pixels.
• No comparison with SRResNet upon which the model was built.
• It is not clear how the model would behave on more practical SR problem, given that $8 \times 8$ to $32 \times 32$ is unrealistic scenario.
• Separate training on rather homogenous data sets (Multi-modality issue).
• In SR problem modelling, LR is not merely a down-sampled version of the normal image, one should take warping, blurring and noise into account, in addition to down-sampling.
References


References


• http://www.intelligent-aerospace.com/articles/2014/05/eiast-sat.html

• http://people.duke.edu/~sf59/
Questions?
Thank you