Reconstruction of High Resolution Images Using Deep Learning

Zhipeng ZHU, Chinese University of Hong Kong
Tianxiang GAO, City University of Hong Kong
Background

- In the field of digital image processing, High-resolution images or videos are commonly needed; but in many cases, people could only obtain low-resolution images.
- Image Super Resolution is a class of techniques that turn a low-resolution image into a high-resolution one for further analysis and processing.
- High-resolution images can provide more information for human interpretation, and improve the quality of automatic machine processing.
Applications

Medical diagnosis

surveillance
Applications

Regular video super-resolution

Astronomy photo super-resolution
Objective and Steps

- Test and compare current deep-learning based super-resolution (SR) models, including single-image SR models and video SR models.
- Improve current SR models using the method of transfer learning.
- Implement our model on Magma DNN
Convolutional Neural Network (CNN) is similar to ordinary neural networks. But the inputs of CNN are images, which enable us to encode certain properties into the architecture. In CNN, we add a particular type of layer, called convolutional layer, to extract features of images.
Key Point -- Transfer Learning

Drawbacks of training an entire neural network:

Sometimes difficult to obtain a dataset of sufficient size;

Training an entire network on a large dataset is very time-consuming, sometimes takes even several weeks.

Important fact:

Pretrained neural network on a large dataset is usually a good feature extractor, thus can be reused in similar tasks.
Models

SRCNN model -- for single-image Super-Resolution
Transfer Learning -- fine-tuned parameter + self-trained layers
3D SRnet model -- for video Super-Resolution
Transfer Learning -- fine-tuned parameter + 3D SRnet
SRCNN model -- for single-image SR

Feature Extraction

Non-linear Mapping

Reconstruction
SRCNN model -- with fine-tuned parameters

Pass the First Convolutional Layer
- Resize by Bicubic
- Filters = 64

Pass the Second Convolutional Layer
- (9 x 9)
- Filters = 32
- Load fine-tuned parameters

Pass the Third Convolutional Layer
- (5 x 5)
- Filters = 1
- Load fine-tuned parameters
- Single Pixel 1
- Single Pixel 2
- Single Pixel 3
- Single Pixel 4
Transfer Learning -- fine-tuned parameter + self-trained layers
SRCNN -- add non-linearity
Transfer Learning -- fine-tuned parameter + non-linearity
<table>
<thead>
<tr>
<th>Performance</th>
<th>Raw Data</th>
<th>Self-Trained SRCNN model</th>
<th>SRCNN model with fine-tuned parameters</th>
<th>SRCNN with transfer-learning (1+2) model</th>
<th>Self-trained SRCNN with added non-linearity</th>
<th>SRCNN with transfer-learning (1+3) model</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR (Peak signal-to-noise ratio)</td>
<td>33.24609 81322429 9</td>
<td>33.9774122 589296</td>
<td>34.39533863 298893</td>
<td>34.568111435 612344</td>
<td>31.75340039 322545</td>
<td>34.420866851 485926</td>
</tr>
<tr>
<td>SSIM (structural similarity index)</td>
<td>0.912191 61971856 61</td>
<td>0.92989156 33331887</td>
<td>0.931424704 3105195</td>
<td>0.9339550069 793681</td>
<td>0.927639833 6420119</td>
<td>0.9340359732 007651</td>
</tr>
<tr>
<td>MAE (Mean Absolute Error)</td>
<td>0.016739 55861304 3767</td>
<td>0.01597587 670676589</td>
<td>0.015128582 491306834</td>
<td>0.0148824250 23144312</td>
<td>0.021569695 539644516</td>
<td>0.0152505775 26577434</td>
</tr>
<tr>
<td>MSE (Mean Squared Error)</td>
<td>0.000931 39330134 29778</td>
<td>0.00076756 489539201 49</td>
<td>0.000720391 6113815921</td>
<td>0.0006896958 805634528</td>
<td>0.000979942 1850334928</td>
<td>0.0007280809 308989662</td>
</tr>
</tbody>
</table>
3D SRnet model -- for video SR

5 single images

Time = -2
Time = 0
Time = +2

1 package of images (according to time = 0 frame)

Convolute without Padding
Depth: 5 → 3

Process the single image in normal 2D conv model

Convolute without Padding
Depth: 3 → 1
### 3D SRnet -- for sequential images

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output Shape</th>
<th>Param #</th>
<th>Padding</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv3d_1 (Conv3D)</td>
<td>(None, 5, 64, 64, 32)</td>
<td>896</td>
<td>With Padding</td>
</tr>
<tr>
<td>conv3d_2 (Conv3D)</td>
<td>(None, 5, 64, 64, 32)</td>
<td>27680</td>
<td>With Padding</td>
</tr>
<tr>
<td>conv3d_3 (Conv3D)</td>
<td>(None, 5, 64, 64, 32)</td>
<td>27680</td>
<td>With Padding</td>
</tr>
<tr>
<td>conv3d_4 (Conv3D)</td>
<td>(None, 3, 62, 62, 32)</td>
<td>27680</td>
<td>Without Padding</td>
</tr>
<tr>
<td>conv3d_5 (Conv3D)</td>
<td>(None, 1, 60, 60, 32)</td>
<td>27680</td>
<td>Without Padding</td>
</tr>
<tr>
<td>reshape_1 (Reshape)</td>
<td>(None, 60, 60, 32)</td>
<td>0</td>
<td>Reshape for 2d model</td>
</tr>
<tr>
<td>conv2d_1 (Conv2D)</td>
<td>(None, 60, 60, 1)</td>
<td>1156</td>
<td>With Padding</td>
</tr>
</tbody>
</table>

Total params: 112,772
Transfer Learning -- fine-tuned parameter + 3D SRnet

Pass the First Convolutional Layer

Pass the Second Convolutional Layer

Load fine-tuned parameters

Filters = 32

Filters = 64

Resized Image 64 x 64

Resize by Bi-cubic

Single Pixel

Convolute without Padding: Depth 3 > 1

Time = 0

Time = 2

Time = 4

Process the single image in normal 2D conv model

Pass 3D Layers without padding

Pass the Last 2D Convolutional Layer

Single Pixel

Single Pixel 1

Single Pixel 2

Single Pixel 3

Single Pixel 4

Filters = 1
Conclusion

Pre-process dataset;
Build and test typical models;
Combine typical models by using transfer learning.

Implement transfer learning with 3D Model;
Use different dataset;
Fine tune hyper-parameters.
Try MagmaDNN for implementation.
Reference

- Li FF, Karpathy A, Johnson J. Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition.
Q & A