Bidirectional Recurrent Convolutional Networks for Video Super-Resolution

Qi Zhang & Yan Huang

Center for Research on Intelligent Perception and Computing (CRIPAC)
National Laboratory of Pattern Recognition (NLPR)
Institute of Automation, Chinese Academy of Sciences (CASIA)

May 10, 2017
CRIPAC mainly focuses on the following research topics related to national public security.

- Biometrics
- Image and Video Analysis
- Big Data and Multi-modal Computing
- Content Security and Authentication
- Sensing and Information Acquisition

CRIPAC receives regular fundings from various Government departments or agencies. It is also supported by funds of R&D projects from many other national and international sources. CRIPAC members publish widely in leading national and international journals and conferences such as IEEE Transactions on PAMI, IEEE Transactions on Image Processing, International Journal of Computer Vision, Pattern Recognition, Pattern Recognition Letters, ICCV, ECCV, CVPR, ACCV, ICPR, ICIP, etc.

http://cripac.ia.ac.cn/en/EN/volumn/home.shtml
Artificial Intelligence Laboratory

Researches on artificial intelligence and deep learning
Outline

1/ Deep Learning
2/ Recurrent Convolutional Networks
3/ Application to Video Super-Resolution
4/ Future Work
Outline

1/ Deep Learning

2/ Recurrent Convolutional Networks

3/ Application to Video Super-Resolution

4/ Future Work
Deep Neural Networks (DNN)

• **Originate from:**
  - 1962 – simple/complex cell, *Hubel and Wiesel*
  - 1970 – efficient error backpropagation, *Linnainmaa*
  - 1979 – deep neocognitron, convolution, *Fukushima*
  - 1987 – autoencoder, *Ballard*
  - 1989 – backpropagation for CNN, *Lecun*
  - 1991 – deep recurrent neural network, *Schmidhuber*
  - 1997 – supervised LSTM RNN, *Schmidhuber*

• **Two drawbacks:**
  - Large numbers of parameters → High computational cost
  - Small training set → Over-fitting problem
Two Recent Developments

**Big Data**

- **ImageNet** Large Scale Visual Recognition Challenge (ILSVRC) 2010-2013

  - 20 object classes: 22,591 images
  - 200 object classes: 456,191 images
  - 1,431,167 images

- **Video surveillance data size (PB)**

<table>
<thead>
<tr>
<th>Year</th>
<th>2009</th>
<th>2010</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10000</td>
<td>13000</td>
<td>18200</td>
<td>27300</td>
<td>54600</td>
<td>87360</td>
</tr>
</tbody>
</table>

**Cheap Computation**

<table>
<thead>
<tr>
<th>Features</th>
<th>Tesla K40</th>
<th>Tesla K20X</th>
<th>Tesla K20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number and Type of GPU</td>
<td>1 Kepler GK110B</td>
<td>1 Kepler GK110</td>
<td></td>
</tr>
<tr>
<td>Peak double precision floating point performance</td>
<td>1.43 Tflops</td>
<td>1.31 Tflops</td>
<td>1.17 Tflops</td>
</tr>
<tr>
<td>Peak single precision floating point performance</td>
<td>4.29 Tflops</td>
<td>3.95 Tflops</td>
<td>3.52 Tflops</td>
</tr>
<tr>
<td>Memory bandwidth (ECC off)</td>
<td>288 GB/sec</td>
<td>250 GB/sec</td>
<td>208 GB/sec</td>
</tr>
<tr>
<td>Memory size (GDDR5)</td>
<td>12 GB</td>
<td>6 GB</td>
<td>5 GB</td>
</tr>
<tr>
<td>CUDA cores</td>
<td>2880</td>
<td>2688</td>
<td>2496</td>
</tr>
</tbody>
</table>

DNN can thus be fitted efficiently
Deep Learning: The Resurgence of DNN

Breakthrough in 2006

Reduction of Dimensionality of Data with Neural Networks

ImageNet: 74% vs. 85%

RNN for sequence analysis

Activity recognition, CVPR2015

Video caption, CVPR2015

Deep Learning promotes the fast development of various visual computing areas

Representation learning

Deep Belief Network

CNN for visual tasks

DeepFace, CVPR2014

RCNN for detection, CVPR2014
Outline

1/ Deep Learning
2/ Recurrent Convolutional Networks
3/ Application to Video Super-Resolution
4/ Future Work
Deep Neural Networks (DNN)

\[ y = \sigma(hW) \]

- \( \mathbf{x} \in \mathbb{R}^d, \mathbf{h} \in \mathbb{R}^n, \mathbf{W} \in \mathbb{R}^{d \times n} \)

- \( h = \sigma(xW), \quad \sigma(t) = \frac{1}{1+e^{-t}} \)

Sigmoid function \( \sigma(t) \)
Recurrent Neural Networks (RNN)

\[ y = \sigma(x_t W + h_{t-1} U) \]

- \( x_t \in \mathbb{R}^d \), \( h_t \in \mathbb{R}^n \), \( W \in \mathbb{R}^{d \times n} \), \( U \in \mathbb{R}^{n \times n} \)

Temporal dependency modeling
Recurrent Convolutional Networks (RCN)

DNN: Deep Neural Networks  
RNN: Recurrent Neural Networks  
CNN: Convolutional Neural Networks
Applications of RCN

- **Video SR**, *NIPS15 & TPAMI17*
- **Scene Labeling**, *NIPS15*
- **Weather Nowcasting**, *NIPS15*
- **Action Recognition**, *ICLR15*
- **Object Recognition**, *CVPR15*
- **Person ReID**, *CVPR16*
1/ Deep Learning

2/ Recurrent Convolutional Networks

3/ Application to Video Super-Resolution

4/ Future Work
Video Super-Resolution

High-resolution devices

Display

Low-resolution videos

High-resolution videos

Super-resolution: denoising, deblurring, upscaling

A great need for super resolving low-resolution videos
Two Main Approaches (1/2)


One-to-One scheme, super resolve each video frame independently

Ignore the intrinsic temporal dependency relation of video frames

Low computational complexity, fast

Two Main Approaches (2/2)

2. Multi-Frame super-resolution [7-11]

Many-to-One scheme, use multiple adjacent frames to super resolve a frame

Model the temporal dependency relation by motion estimation

High computational complexity, slow

Motivation

RNN: Recurrent Neural Networks
SR: Super-Resolution

- RNN can model **long-term contextual information** of temporal sequences well
- Convolutional operation can **scale to full videos** of any spatial size and temporal step

➢ Propose **bidirectional recurrent convolutional networks**, different from vanilla RNN:

1. Commonly-used full connections are replaced with weight-sharing convolutions
2. Conditional convolutions are added for learning visual-temporal dependency relation
Bidirectional Recurrent Convolutional Networks

- **Feedforward convolution**: learn spatial dependency between a low-resolution frame and its high-resolution result.

- **Recurrent convolution**: model long-term temporal dependency relation across video frames.

- **Conditional convolution**: enhance visual-temporal dependency relation modeling.

Diagram:

- Input layer (low-resolution frame)
- First hidden layer
- Second hidden layer
- Output layer (high-resolution frame)
- Second hidden layer
- First hidden layer
- Input layer (low-resolution frame)
Learning

- Define an end-to-end mapping $O(\cdot)$ from low-resolution frames $X$ to high-resolution frames $Y$
- Learning proceeds by optimizing the Mean Square Error (MSE) between predicted frames $O(X)$ and $Y$

$$L = \| O(X) - Y \|^2$$

- stochastic gradient descent
- small learning rate in the output layer: $1e-4$
Experiments

• Train the model on 25 YUV format video sequences
  – volume-based training
  – number of volumes: roughly 41,000
  – volume size: $32 \times 32 \times 10$

• Test on a variety of real world videos
  – severe motion blur
  – motion aliasing
  – complex motions
Table 1: The results of PSNR (dB) and test time (sec) on the test video sequences.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>Time</td>
<td>PSNR</td>
<td>Time</td>
<td>PSNR</td>
</tr>
<tr>
<td>Dancing</td>
<td>26.83</td>
<td>-</td>
<td>26.80</td>
<td>45.47</td>
<td>27.69</td>
</tr>
<tr>
<td>Flag</td>
<td>26.35</td>
<td>-</td>
<td>26.28</td>
<td>12.89</td>
<td>27.61</td>
</tr>
<tr>
<td>Fan</td>
<td>31.94</td>
<td>-</td>
<td>32.50</td>
<td>12.92</td>
<td>33.55</td>
</tr>
<tr>
<td>Treadmill</td>
<td>21.15</td>
<td>-</td>
<td>21.27</td>
<td>15.47</td>
<td>22.22</td>
</tr>
<tr>
<td>Turbine</td>
<td>25.09</td>
<td>-</td>
<td>25.77</td>
<td>16.49</td>
<td>27.00</td>
</tr>
<tr>
<td>Average</td>
<td>26.27</td>
<td>-</td>
<td>26.52</td>
<td>20.64</td>
<td>27.61</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>Time</td>
<td>PSNR</td>
<td>Time</td>
<td>PSNR</td>
</tr>
<tr>
<td>Dancing</td>
<td>27.64</td>
<td>4.20</td>
<td>27.81</td>
<td>1.41</td>
<td>-</td>
</tr>
<tr>
<td>Flag</td>
<td>27.48</td>
<td>0.96</td>
<td>28.04</td>
<td>0.36</td>
<td>-</td>
</tr>
<tr>
<td>Fan</td>
<td>33.46</td>
<td>1.76</td>
<td>33.61</td>
<td>0.60</td>
<td>-</td>
</tr>
<tr>
<td>Treadmill</td>
<td>22.22</td>
<td>0.57</td>
<td>22.42</td>
<td>0.15</td>
<td>-</td>
</tr>
<tr>
<td>Turbine</td>
<td>26.98</td>
<td>0.80</td>
<td>27.50</td>
<td>0.23</td>
<td>-</td>
</tr>
<tr>
<td>Average</td>
<td>27.52</td>
<td>1.66</td>
<td>27.87</td>
<td>0.55</td>
<td>-</td>
</tr>
</tbody>
</table>

Surpass state-of-the-art methods in PSNR, due to the effective temporal dependency modelling.
Model Architecture

- Investigate the impact of our model architecture on the performance
- Take a simplified network containing only feedforward ($v$) convolution as a benchmark
- Study its variants by successively adding the bidirectional ($b$), recurrent ($r$) and conditional ($t$) schemes

Table 1: The results of PSNR (dB) by variants of BRCN on the testing video sequences.
Running Time

Figure: Speed vs. PSNR for all the comparison methods.

- Outperform both single-image and multi-frame SR methods
- Achieve comparable speed with the fastest single-image SR methods
Closeup Comparison

Figure: Comparison among original frames (2th, 3th and 4th frames, from the top row to the bottom) of the Dancing video and super resolved results by Bicubic, 3DSKR, ANR and BRCN, respectively. Our method is able to recover more image details than others, under severe motion conditions.
Example

Upscaling factor: 4
\[87 \times 157 \rightarrow 348 \times 628\]

Comparison:
- Bicubic (top)
- Ours (bottom)
Conclusion

• Bidirectional Recurrent Convolutional Networks
  – bidirectional recurrent and conditional convolutions
  – an end-to-end framework, without pre/post-processing
  – well performance and fast speed

For more details, please refer to the following papers:


Outline

1. Deep Learning
2. Recurrent Convolutional Networks
3. Application to Video Super-Resolution
4. Future Work
Future Work

➢ For performance improvement
   – extend our model to have a deeper architecture, e.g., based on 19 layers VGG net
   – incorporate some effective strategies, e.g., motion ensemble and residual connection

➢ For speed acceleration
   – replace the used pre-upsampling by learning diverse upsampling filters with deconvolution layers

➢ Others
   – collect a large-scale high-resolution video dataset, and try to learn our model directly from raw videos
Acknowledgement

NVAIL
Artificial Intelligence Laboratory
Sponsor excellent hardware resources
THANK YOU

yhuang@nlpr.ia.ac.cn