Single Image Super-Resolution

Author: Mikuláš Bankovič

Faculty of Informatics, Masaryk University

October 22, 2020

Introduction

Motivation

Why Super-Resolution (SR)? Link

Introduction

Motivation

Why SR? Link

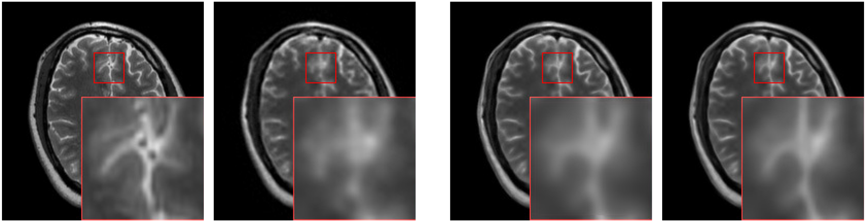
- Games

Introduction

Motivation

Why SR? Link

- Games
- Medical imaging



(a) Ground-truth (b) SCSR[13] (c) ResNet (d) DGGRN

Introduction

Motivation

Why SR? Link

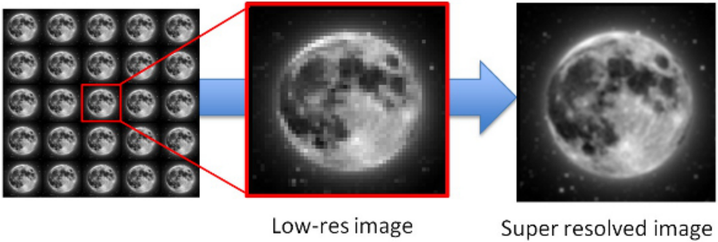
- Games
- Medical imaging
- Photography details

Introduction

Motivation

Why SR? Link

- Games
- Medical imaging
- Photography details
- Astronomy



Introduction

Motivation

Why SR? Link

- Games
- Medical imaging
- Photography details
- Astronomy
- Face and character recognition

Introduction

Motivation

Why SR? Link

- Games
- Medical imaging
- Photography details
- Astronomy
- Face and character recognition
- Project video699[3]

Introduction

Motivation

Why SR? Link

- Games
- Medical imaging
- Photography details
- Astronomy
- Face and character recognition
- Project video699[3]
- Super-scaling of FFFI movies

Metrics

- The Peek Signal Noise Ratio (PSNR) (in dB) is defined as following:

$$PSNR = 10 \times \log_{10}\left(\frac{MAX_I^2}{MSE}\right),$$

where MAX_I is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255.

Metrics

- The PSNR (in dB) is defined as following:

$$PSNR = 10 \times \log_{10}\left(\frac{MAX_I^2}{MSE}\right),$$

where MAX_I is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255.

- SSIM - weighted combination of luminance, contrast and structure:

$$\mathrm{SSIM}(x,y) = \left[l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma}\right]$$

Metrics

- The PSNR (in dB) is defined as following:

$$PSNR = 10 \times \log_{10}\left(\frac{MAX_I^2}{MSE}\right),$$

  where MAX_I is the maximum possible pixel value of the image. When the pixels are represented using 8 bits per sample, this is 255.
- SSIM - weighted combination of luminance, contrast and structure:
- MOS - mostly human opinions on 5 number scale

History of SR

Bilinear and bicubic interpolation:

- no prior knowledge about images
- no way to fine-tune to specific dataset
- does not improve with more data

Sparse-coding-based methods:

- The methods are part of example-based learning methods.
- They consist of a multiple-step pipeline:
  1. Crop overlapping patches and preprocess them (substract mean and normalize)
  2. Encode these patches by Low-Resolution (LR) dictionary
  3. Encoded coefficients are passed to the High-Resolution (HR) dictionary
  4. Overlapping HR patches are aggregated
- Focus on optimizing and improving dictionaries with mapping, while disregarding other steps.
- They often have to solve optimization problems on inference.

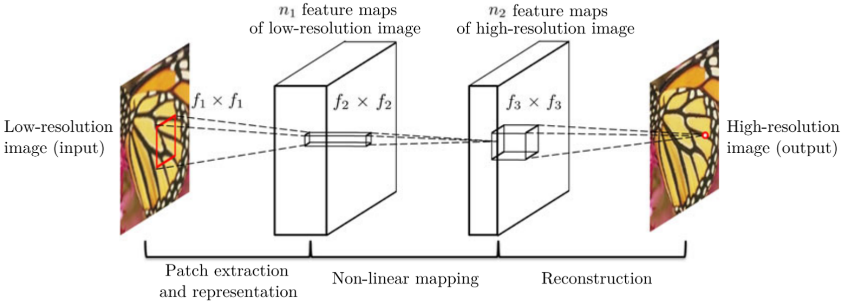
Super-Resolution Convolutional Neural Network (SRCNN)

- Given by Dong et al. [1], the Convolutional Neural Network (CNN) is equivalent to the previous pipeline.
- That brings multiple advantages:
  - The inference consists only from feed-forward pass.
  - The pipeline is unified, therefore, each step is optimized during training.
  - The dictionaries are not explicitly formed, but included in the weights.
  - Provides superior quality and speed performance (Next slides).

SRCNN

- SRCNN is a simple CNN with three convolutional layers.
- The input image is firstly upscaled using bicubic interpolation.
- The next step is feed-forward pass through the network.
- The different architecture involved changes in filter size: 9-1-5, 9-5-5, 11-5-7, etc.

SRCNN


n_1 feature maps
of low-resolution image
n_2 feature maps
of high-resolution image
$f_1 \times f_1$
$f_2 \times f_2$
$f_3 \times f_3$
Low-resolution
image (input)
High-resolution
image (output)
Patch extraction
and representation
Non-linear mapping
Reconstruction

SRCNN

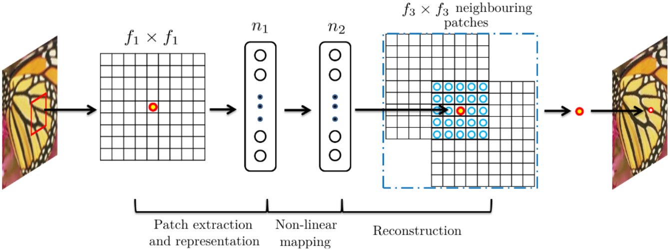
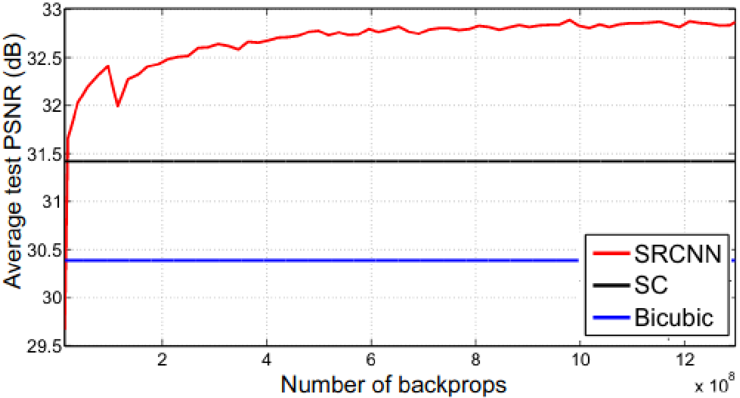


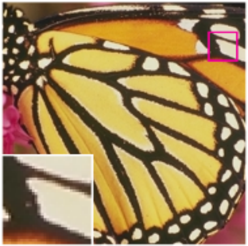
Fig. 3. An illustration of sparse-coding-based methods in the view of a convolutional neural network.

SRCNN

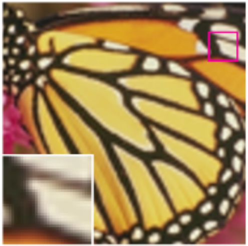


Average test PSNR (dB)
33
32.5
32
31.5
31
30.5
30
29.5
2
4
6
8
10
12
Number of backprops
x 10^8
SRCNN
SC
Bicubic

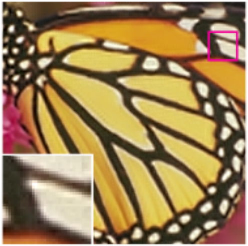
SRCNN



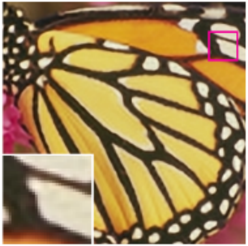
Original / PSNR



Bicubic / 24.04 dB

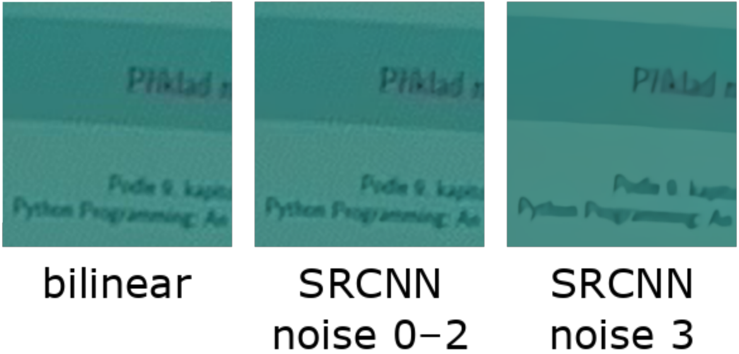


SC / 25.58 dB



SRCNN / 27.95 dB

SRCNN



Problems

- The bicubic interpolation is an expensive operation that often introduce side-effects as blurring or noise amplification.
- More data could overfit the network, because its smaller size.
- Most of the operations are performed in an expensive HR space.

FSRCNN

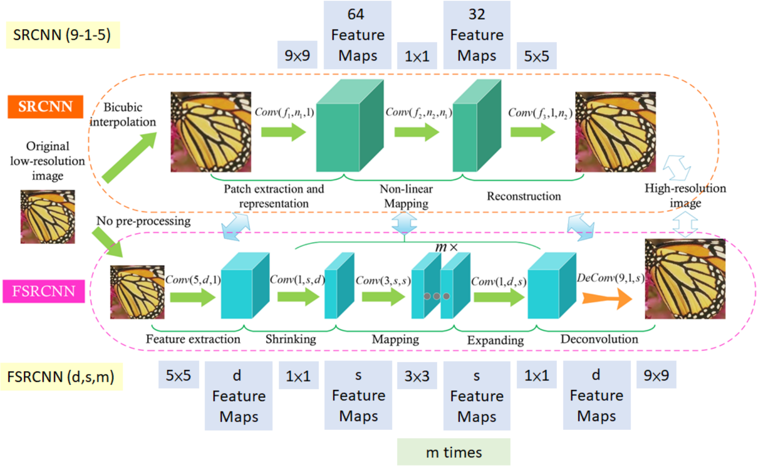
The main differences between SRCNN and Fast Super-Resolution Convolutional Neural Network (FSRCNN):

- There is no pre-processing or upsampling at the beginning. The feature extraction took place in the LR space.
- A 1×1 convolution is used after the initial 5×5 convolution to reduce the number of channels, and hence lesser computation and memory, similar to how the Inception[4] network is developed.
- Multiple 3×3 convolutions are used, instead of having a big convolutional filter, similar to how the VGG network works by simplifying the architecture to reduce the number of parameters.
- Upsampling is done by using a learnt transposed convolution, thus improving the model.

FSRCNN

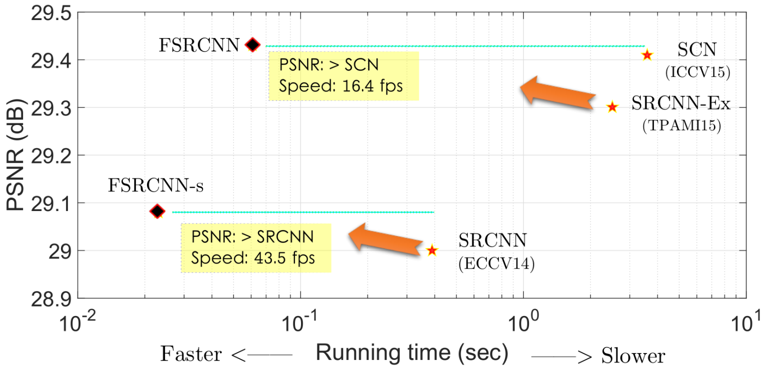
- 17.36 times faster than SRCNN and can run in real time (24 fps) with a generic CPU.
- All layers except from the last can be shared with multiple upscaling factors.
- Transposed convolution LINK[2]
- Transposed convolution with stride LINK[2]

FSRCNN



SRCNN (9-1-5)
9x9
64
Feature
Maps
1x1
32
Feature
Maps
5x5
SRCNN
Bicubic interpolation
Original low-resolution image
$Conv(f_1, n_1, 1)$
$Conv(f_2, n_2, n_1)$
$Conv(f_3, 1, n_2)$
Patch extraction and representation
Non-linear Mapping
Reconstruction
High-resolution image
No pre-processing
$m\times$
FSRCNN
$Conv(5, d, 1)$
$Conv(1, s, d)$
$Conv(3, s, s)$
$Conv(1, d, s)$
$DeConv(9, 1, s)$
Feature extraction
Shrinking
Mapping
Expanding
Deconvolution
FSRCNN (d,s,m)
5x5
d
Feature
Maps
1x1
s
Feature
Maps
3x3
s
Feature
Maps
1x1
d
Feature
Maps
9x9
m times

FSRCNN



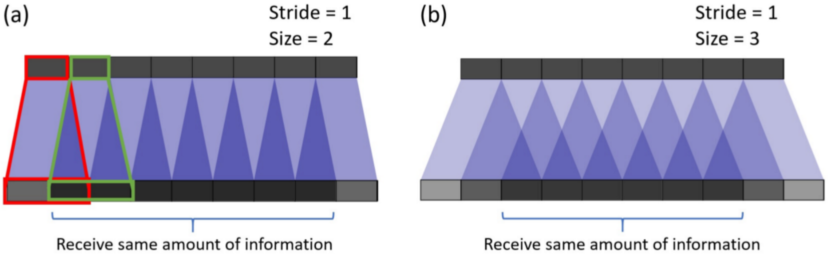
29.5
29.4
29.3
29.2
29.1
29
28.9
PSNR (dB)
FSRCNN
PSNR: > SCN
Speed: 16.4 fps
SCN
(ICCV15)
SRCNN-Ex
(TPAMI15)
FSRCNN-s
PSNR: > SRCNN
Speed: 43.5 fps
SRCNN
(ECCV14)
10^-2
10^-1
10^0
10^1
Faster <—— Running time (sec) ——> Slower

FSRCNN


Original / PSNR
Bicubic / 24.04 dB
SRF / 27.96 dB
SRCNN / 27.58 dB
SRCNN-Ex / 27.95 dB
SCN / 28.57 dB
FSRCNN-s / 27.73 dB
FSRCNN / **28.68** dB

ESPCN

- Efficient Sub-Pixel Convolutional Neural network (ESPCN) introduces the concept of sub-pixel convolution to replace the transposed convolution layer for upsampling. This solves two problems associated with it:
  1. Transposed convolution happens in the high resolution space, and thus is more computationally expensive.
  2. It resolves the checkerboard issue in deconvolution, which occurs due to the overlap operation of convolution.



Sub-pixel convolutional layers

- In the recent literature they are called pixel-shuffle or depth-to-space layers.
- Pixels from multiple channels in a low resolution image are rearranged to a single channel in a high resolution image.

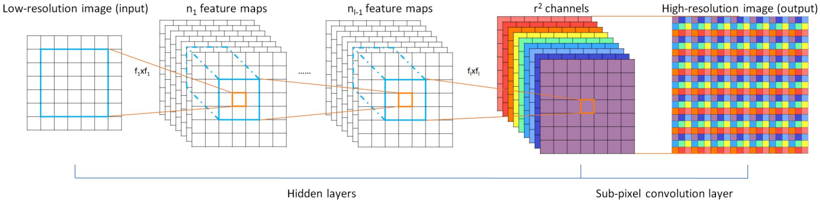
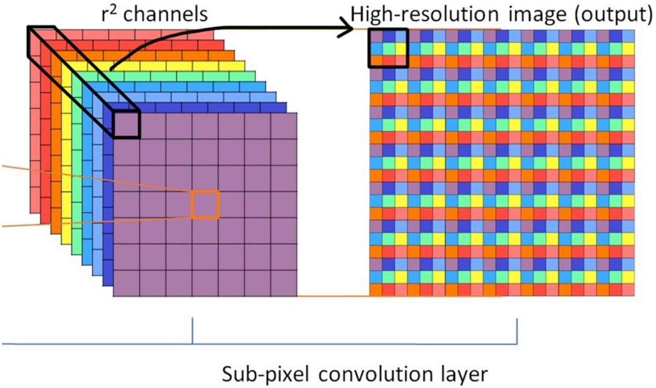


Figure 1. The proposed efficient sub-pixel convolutional neural network (ESPCN), with two convolution layers for feature maps extraction, and a sub-pixel convolution layer that aggregates the feature maps from LR space and builds the SR image in a single step.

Sub-pixel convolutional layers

- In the recent literature they are called pixel-shuffle or depth-to-space layers.
- Pixels from multiple channels in a low resolution image are rearranged to a single channel in a high resolution image.



ESPCN

Figure: Lancsoz (left) vs ESPCN (right)

Super-Resolution Generative Adversarial Network (SRGAN)

- Problem: All previous methods were train on MSE loss function. However, ideal MSE image is not offhand the most photo-realistic
- Solution: Generative Adversarial Network (GAN)

SRGAN

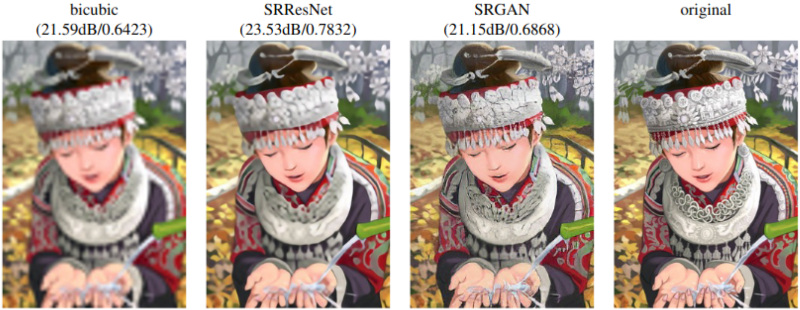


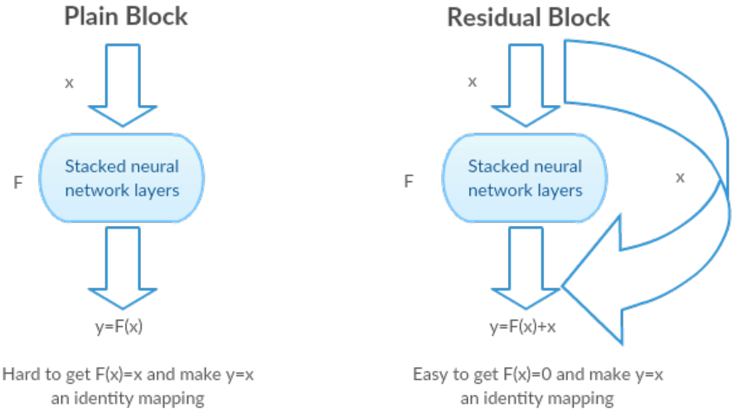
Figure 2: From left to right: bicubic interpolation, deep residual network optimized for MSE, deep residual generative adversarial network optimized for a loss more sensitive to human perception, original HR image. Corresponding PSNR and SSIM are shown in brackets. [4× upscaling]

SRGAN

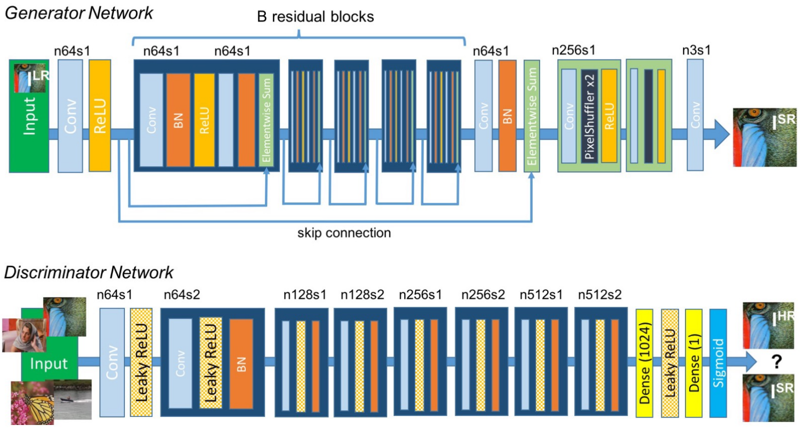
- Problem: All previous methods were train on MSE loss function. However, ideal MSE image is not offhand the most photo-realistic
- Solution: GAN

- Problem: As in other computer vision disciplines, deeper models are more successful, however, harder to train due to some aspects, such as vanishing gradient problem.
- Solution: ResNet

ResNet


Plain Block
x
F
Stacked neural network layers
y=F(x)
Hard to get F(x)=x and make y=x an identity mapping
Residual Block
x
F
Stacked neural network layers
x
y=F(x)+x
Easy to get F(x)=0 and make y=x an identity mapping

GAN


Generator Network
B residual blocks
n64s1
n64s1
n64s1
n64s1
n256s1
n3s1
Input
Conv
ReLU
Conv
BN
ReLU
Elementwise Sum
Conv
BN
Elementwise Sum
Conv
PixelShuffler x2
ReLU
Conv
skip connection
Discriminator Network
n64s1
n64s2
n128s1
n128s2
n256s1
n256s2
n512s1
n512s2
Input
Conv
Leaky ReLU
Conv
Leaky ReLU
BN
Dense (1024)
Leaky ReLU
Dense (1)
Sigmoid
?

SRGAN

- This paper generates State Of The Art (SOTA) results on upsampling (4x) as measured by PSNR and SSIM with 16 block deep SRResNet network optimized for MSE.
- The authors proposed a new SRGAN in which the authors replace the MSE based content loss with the loss calculated on VGG layers.
- SRGAN was able to generate SOTA results which the author validated with extensive MOS test on three public benchmark datasets.
- Use 2 losses for generator network: MSE and function based on the euclidean distance between feature maps extracted from the VGG19 network.

Loss

$$l^{SR}_{MSE} = \frac{1}{r^2WH}\sum_{x=1}^{rW}\sum_{y=1}^{rH}(I^{HR}_{x,y} - G_{\theta_G}(I^{LR})_{x,y})^2$$

$$l^{SR}_{VGG/i.j} = \frac{1}{W_{i,j}H_{i,j}}\sum_{x=1}^{W_{i,j}}\sum_{y=1}^{H_{i,j}}(\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2$$

$$l^{SR}_{Gen} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

$$l^{SR} = \underbrace{\underbrace{l^{SR}_{\mathrm{X}}}_{\text{content loss}} + \underbrace{10^{-3} l^{SR}_{Gen}}_{\text{adversarial loss}}}_{\text{perceptual loss (for VGG based content losses)}}$$

Loss

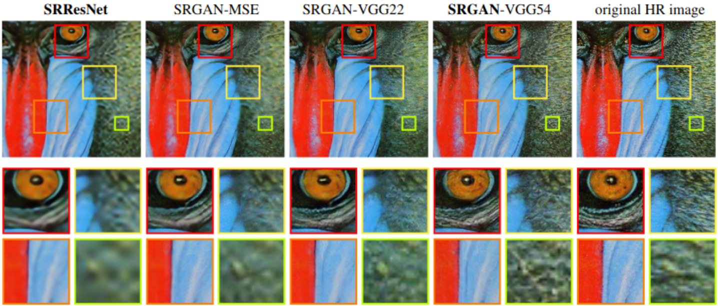
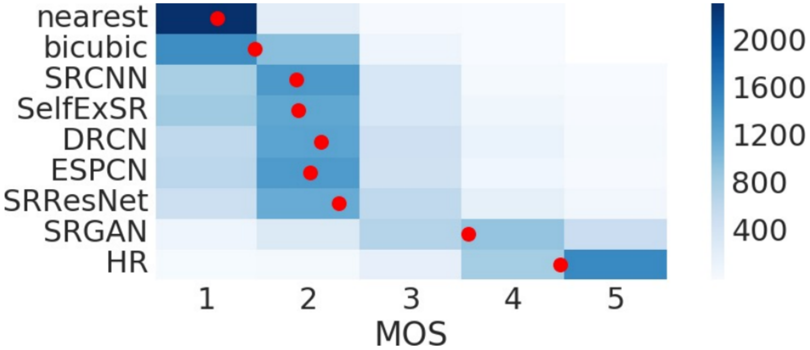


Figure 6: **SRResNet** (left: a,b), SRGAN-MSE (middle left: c,d), SRGAN-VGG2.2 (middle: e,f) and **SRGAN**-VGG54 (middle right: g,h) reconstruction results and corresponding reference HR image (right: i,j). [4× upscaling]

Loss


nearest
bicubic
SRCNN
SelfExSR
DRCN
ESPCN
SRResNet
SRGAN
HR
1
2
3
4
5
MOS
2000
1600
1200
800
400

Loss

Set5	nearest	bicubic	SRCNN	SelfExSR	DRCN	ESPCN	**SRResNet**	**SRGAN**	HR
PSNR	26.26	28.43	30.07	30.33	31.52	30.76	**32.05**	29.40	∞
SSIM	0.7552	0.8211	0.8627	0.872	0.8938	0.8784	**0.9019**	0.8472	1
MOS	1.28	1.97	2.57	2.65	3.26	2.89	3.37	**3.58**	4.32
Set14									
PSNR	24.64	25.99	27.18	27.45	28.02	27.66	**28.49**	26.02	∞
SSIM	0.7100	0.7486	0.7861	0.7972	0.8074	0.8004	**0.8184**	0.7397	1
MOS	1.20	1.80	2.26	2.34	2.84	2.52	2.98	**3.72**	4.32
BSD100									
PSNR	25.02	25.94	26.68	26.83	27.21	27.02	**27.58**	25.16	∞
SSIM	0.6606	0.6935	0.7291	0.7387	0.7493	0.7442	**0.7620**	0.6688	1
MOS	1.11	1.47	1.87	1.89	2.12	2.01	2.29	**3.56**	4.46

Problems

- Expensive training
- A little bit less expensive inference

Future

- Residual networks?
- Attention-based networks?
- GANs?
- Progressive Reconstruction Networks?

Future

TABLE 2
Super-resolution methodology employed by some representative models. The "Fw.", "Up.", "Rec.", "Res.", "Dense.", "Att." represent SR frameworks, upsampling methods, recursive learning, residual learning, dense connections, attention mechanism, respectively.

Method	Publication	Fw.	Up.	Rec.	Res.	Dense	Att.	$\mathcal{L}_{L1}$	$\mathcal{L}_{L2}$	Keywords
SRCNN [22]	2014, ECCV	Pre.	Bicubic						✓	
DRCN [82]	2016, CVPR	Pre.	Bicubic	✓	✓				✓	Recursive layers
FSRCNN [43]	2016, ECCV	Post.	Deconv						✓	Lightweight design
ESPCN [156]	2017, CVPR	Pre.	Sub-Pixel						✓	Sub-pixel
LapSRN [27]	2017, CVPR	Pro.	Bicubic		✓			✓		$\mathcal{L}_{pixel\_Cha}$
DRRN [56]	2017, CVPR	Pre.	Bicubic	✓	✓				✓	Recursive blocks
SRResNet [25]	2017, CVPR	Post.	Sub-Pixel		✓				✓	$\mathcal{L}_{Con.}$, $\mathcal{L}_{TV}$
SRGAN [25]	2017, CVPR	Post.	Sub-Pixel		✓					$\mathcal{L}_{Con.}$, $\mathcal{L}_{GAN}$
EDSR [31]	2017, CVPRW	Post.	Sub-Pixel		✓			✓		Compact and large-size design
EnhanceNet [8]	2017, ICCV	Pre.	Bicubic		✓					$\mathcal{L}_{Con.}$, $\mathcal{L}_{GAN}$, $\mathcal{L}_{texture}$
MemNet [55]	2017, ICCV	Pre.	Bicubic	✓	✓	✓			✓	Memory block
SRDenseNet [79]	2017, ICCV	Post.	Deconv		✓	✓			✓	Dense connections
DBPN [57]	2018, CVPR	Iter.	Deconv		✓	✓			✓	Back-projection
DSRN [85]	2018, CVPR	Pre.	Deconv	✓	✓				✓	Dual state
RDN [93]	2018, CVPR	Post.	Sub-Pixel		✓	✓		✓		Residual dense block
CARN [28]	2018, ECCV	Post.	Sub-Pixel	✓	✓	✓		✓		Cascading
MSRN [99]	2018, ECCV	Post.	Sub-Pixel		✓			✓		Multi-path
RCAN [70]	2018, ECCV	Post.	Sub-Pixel		✓		✓	✓		Channel attention
ESRGAN [103]	2018, ECCVW	Post.	Sub-Pixel		✓	✓		✓		$\mathcal{L}_{Con.}$, $\mathcal{L}_{GAN}$
RNAN [106]	2019, ICLR	Post.	Sub-Pixel		✓		✓	✓		Non-local attention
Meta-RDN [95]	2019, CVPR	Post.	Meta Upscale		✓	✓		✓		Magnification-arbitrary
SAN [105]	2019, CVPR	Post.	Sub-Pixel		✓		✓	✓		Second-order attention
SRFBN [86]	2019, CVPR	Post.	Deconv	✓	✓	✓		✓		Feedback mechanism

Thank You for Your Attention!

Bibliography I

[1] Chao Dong et al. “Image Super-Resolution Using Deep Convolutional Networks”. In: (). URL: `http://arxiv.org/abs/1501.00092` (visited on 10/05/2020).

[2] Vincent Dumoulin and Francesco Visin. *A guide to convolution arithmetic for deep learning*. 2018. URL: `https://arxiv.org/abs/1603.07285v2` (visited on 10/20/2020).

[3] Vít Novotný. *video699. Automatic alignment of lecture recordings with study materials*. 2018. URL: `https://github.com/video699` (visited on 01/10/2020).

Bibliography II

[4] Christian Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: (2015). URL: http://arxiv.org/abs/1512.00567 (visited on 10/20/2020).

CNN Convolutional Neural Network. 14–20

ESPCN Efficient Sub-Pixel Convolutional Neural network. 27, 30

FSRCNN Fast Super-Resolution Convolutional Neural Network. 22–26

GAN Generative Adversarial Network. 31–33

HR High-Resolution. 13, 21

LR Low-Resolution. 13, 22

MOS Mean Opinion Score. 10–12, 36

PSNR Peek Signal Noise Ratio. 10–12, 36

SOTA State Of The Art. 36

SR Super-Resolution. 2–9, 13

MUNI
FACULTY
OF INFORMATICS