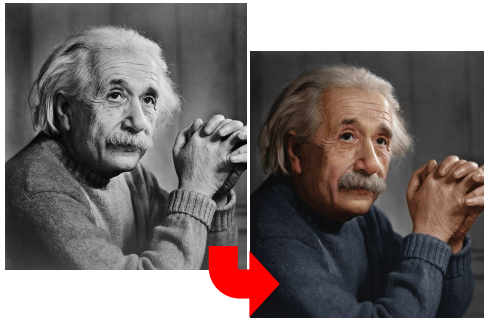
Università degli
Studi di Milano-Bicocca

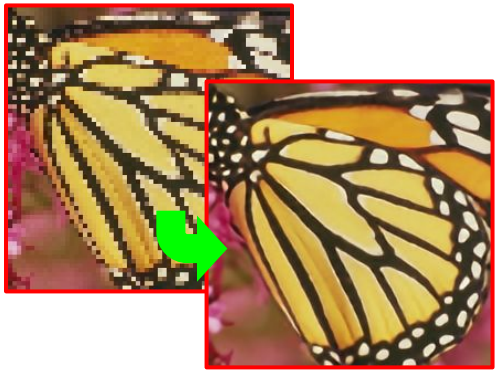
Single Image Super Resolution

Dr. Simone Zini

a.a. 2022-2023

Image processing and Neural Networks

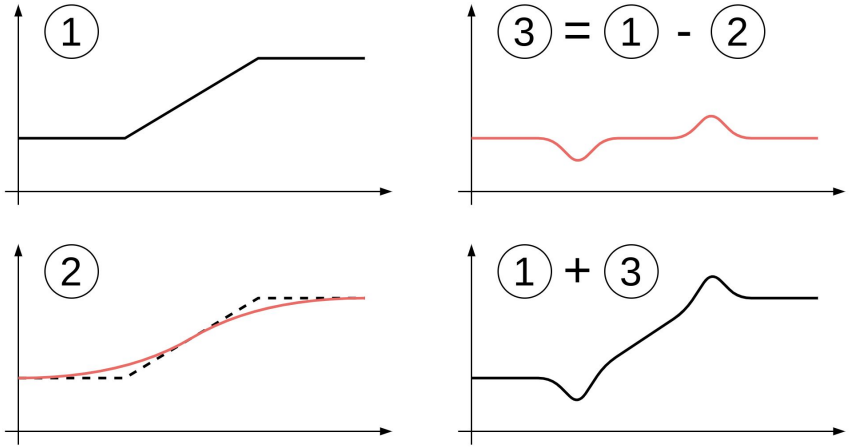


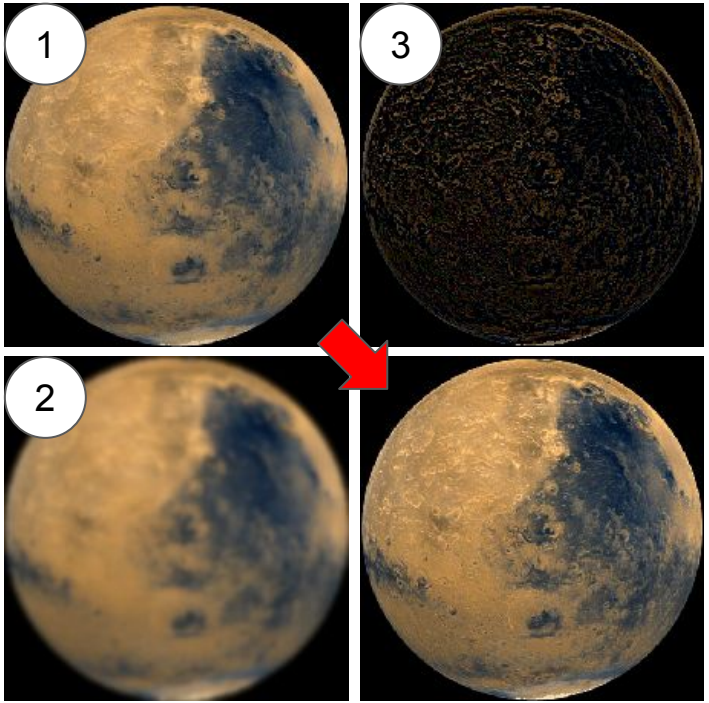


Enhancement of images, in relation to specific tasks, in order to make the images more “readable” by humans and machines.

Traditional approaches

e.g. Unsharp masking





- Based on observations and assumptions
- Handcrafted: may not work in scenarios with different assumptions

Data driven (Machine Learning) approaches

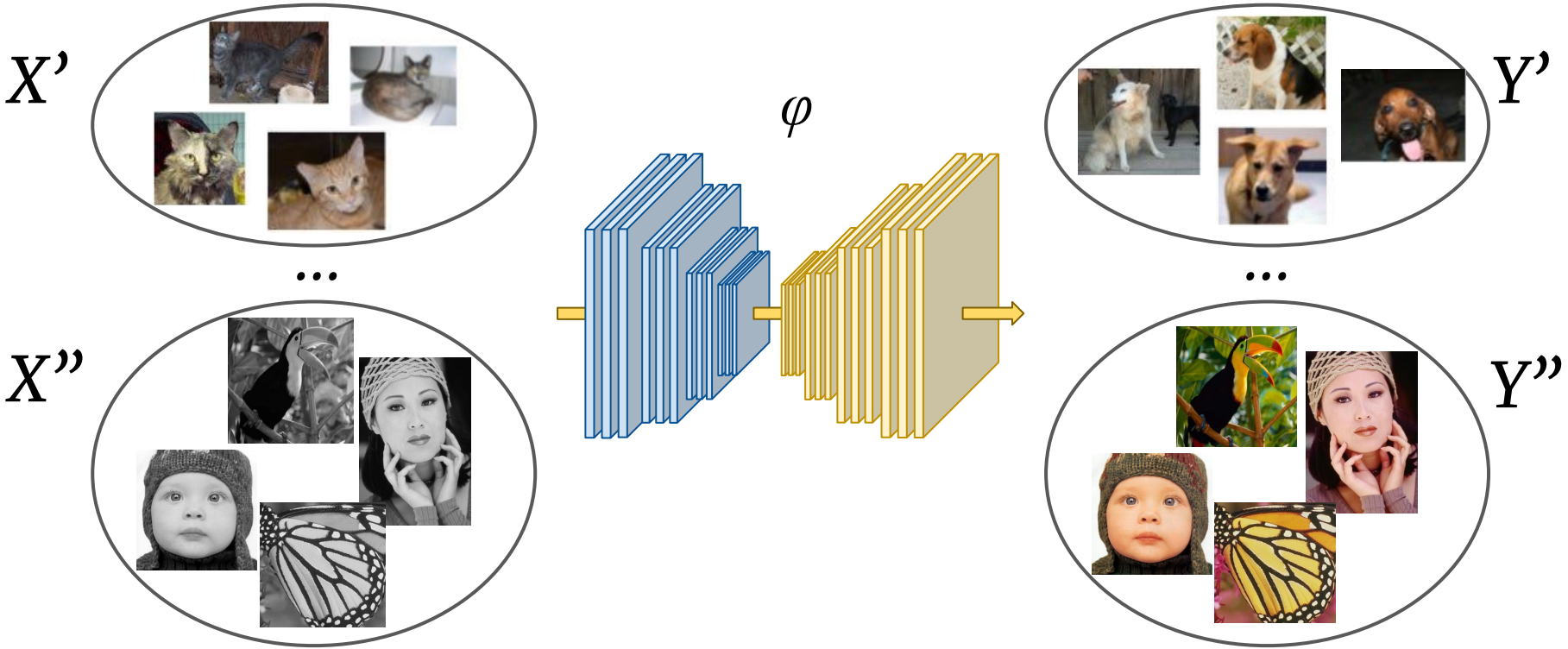
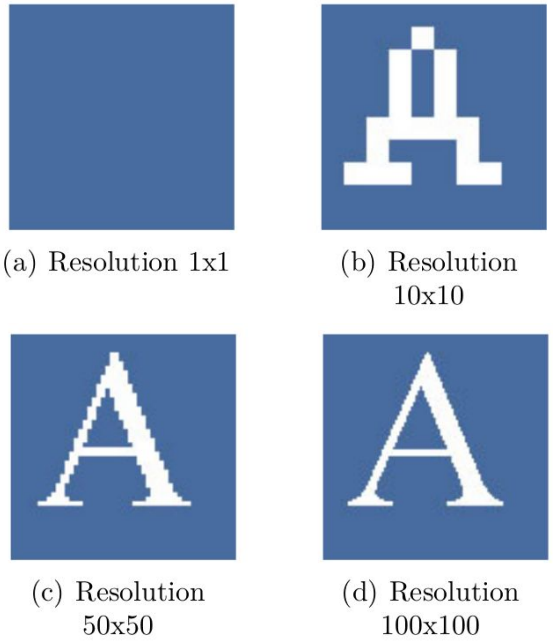


Image to Image translation: defined as the task of translating from one possible representation or style of the scene to another.

Image Resolution and Upsampling

Image spatial resolution



(a) Resolution 1x1

(b) Resolution 10x10

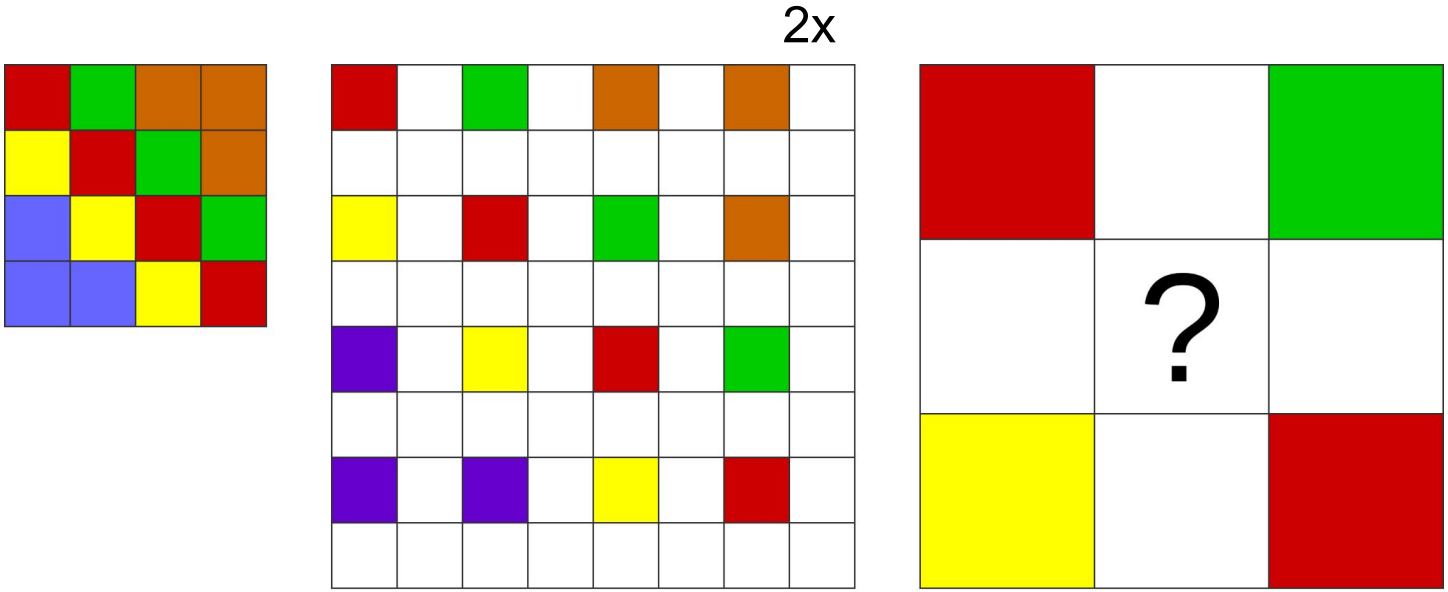
(c) Resolution 50x50

(d) Resolution 100x100

The spatial resolution is defined as the smallest discernible detail in an image.

Operations on image resolution

Image upsampling



Upscaling an image leads to empty pixels that need to be filled.

How?

Image scaling

(a) Original image

(b) Nearest-neighbor

(c) Bilinear

(d) Bicubic

(e) Fourier-based

(f) Edge-direction

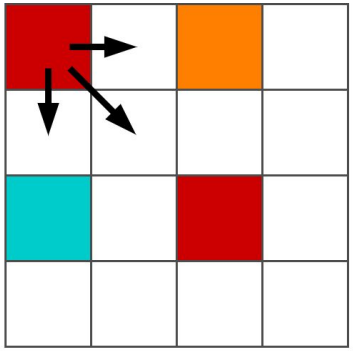
(g) HQX

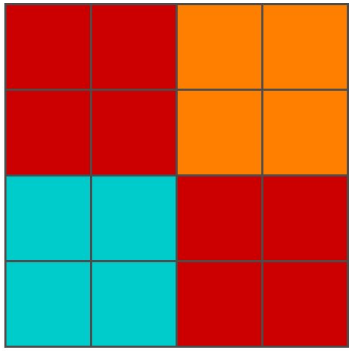
(h) Vectorization

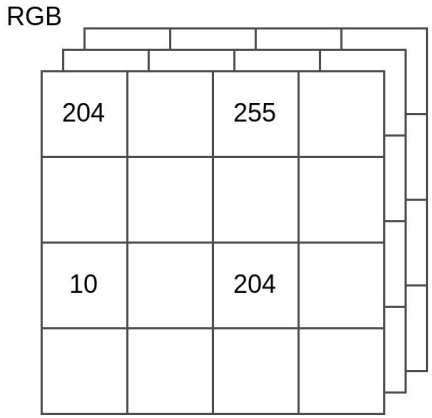
Nearest neighbour upscaling

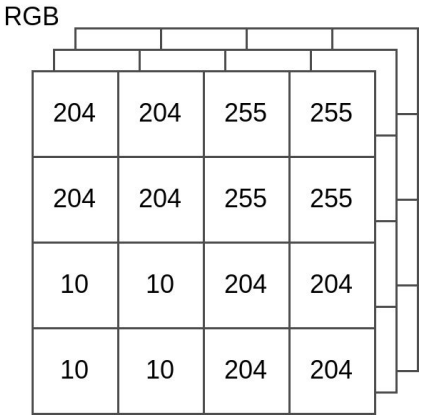
(b)
Nearest-neighbor

- Quick and simple algorithm
- "Blocky" results
- No interpolation in the signal whatsoever









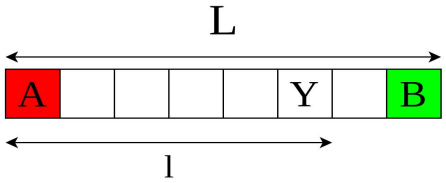
Copy the value in the original nearest pixel

Linear and Bilinear interpolation

(c) Bilinear

- Smoother results wrt Nearest Neighbour
- Color aberrations
- Still blocketization artifacts

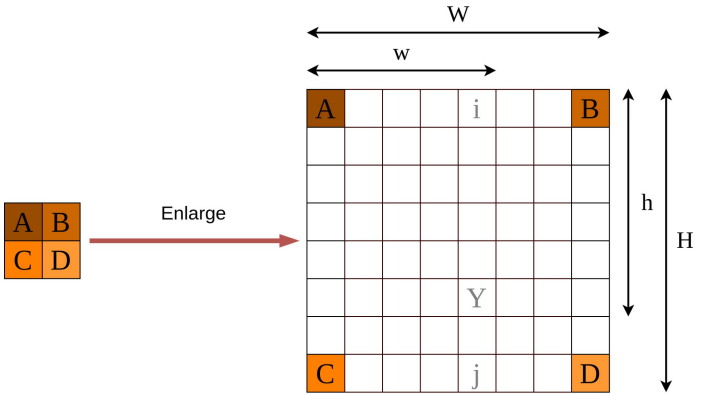
Linear interpolation:



$$\frac{Y-A}{l} = \frac{B-A}{L}$$

$$Y = A + \frac{l(B-A)}{L}$$

Bilinear interpolation (Extension in 2D):

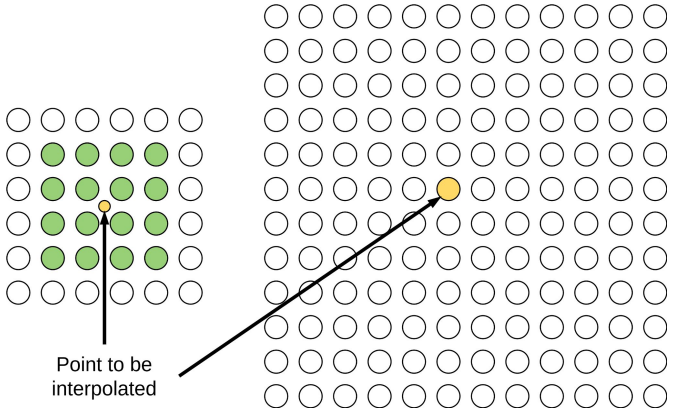


$$Y = A(1-w)(1-h) + B(w)(1-h) + C(h)(1-w) + D(w \cdot h)$$

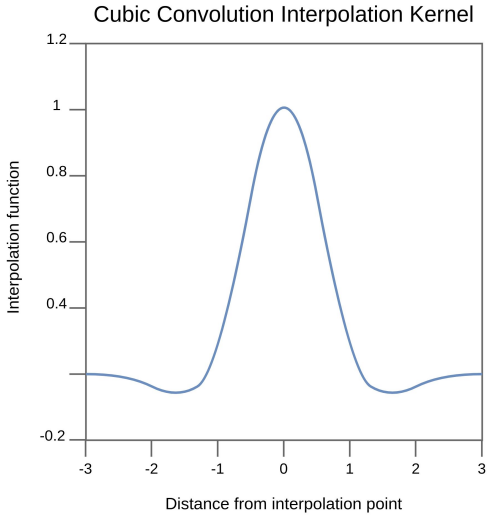
For one directional linear interpolation the number of pixels needed is two; in the case of a bilinear interpolation, the number of grid points needed is four

Bicubic interpolation

(d) Bicubic



16 interpolation points used



$$u(s) = \begin{cases} \frac{3}{2}|s|3 = \frac{5}{2}|s|2 + 1 & 0 <= |s| < 1 \\ -\frac{1}{2}|s|3 + \frac{5}{2}|s|2 - 4|s| + 2 & 1 <= |s| < 2 \\ 0 & 2 < |s| \end{cases}$$

- Smoother results wrt Nearest Neighbour
- Less aberrations than Bilinear
- Images looks “blurry”
- More time needed

$$g(x,y) = \sum_{l=-1}^{2} \sum_{m=-1}^{2} c_{j+l,k+m} u(distance_x) u(distance_y)$$

Super Resolution

Definition of Super Resolution

LR Image

HR Image

Downsampling

Gaussian Kernel

Noise

$$\mathcal{D}(I_{HR};\theta_{\mathcal{D}}) = (I_{HR} \circledast \mathcal{K})\downarrow_s + n$$

Objective: improve the resolution of an imaging system.

- Single-Image Super Resolution
  - Single input Low Resolution source image
- Multi-Image Super Resolution
  - Multiple Low Resolution images of the same scene
  - ↪ Video Super Resolution: Multiple Low Resolution frames

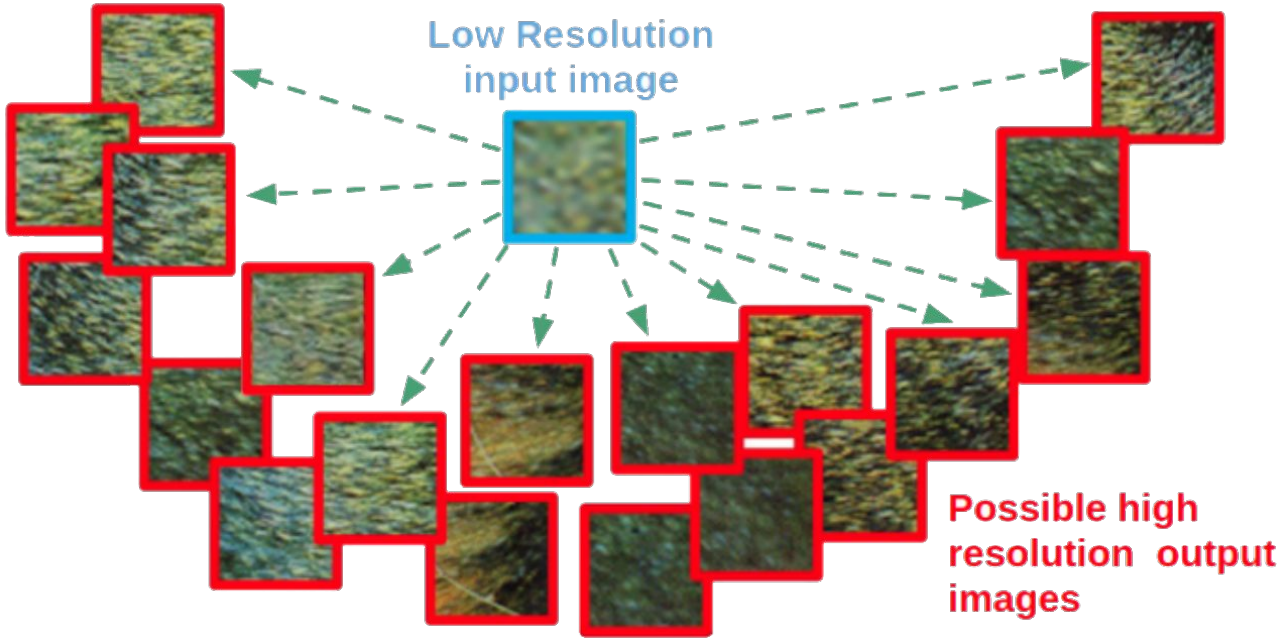
SISR, an ill-posed problem

High-Res image

Low-Res image tile

To which high resolution patch corresponds our low resolution input?

SISR, an ill-posed problem



Multiple possible solutions are **equally** acceptable starting from the same input.

SISR, an ill-posed problem

Multiple possible solutions are **equally** acceptable starting from the same input.

Approaches to SISR

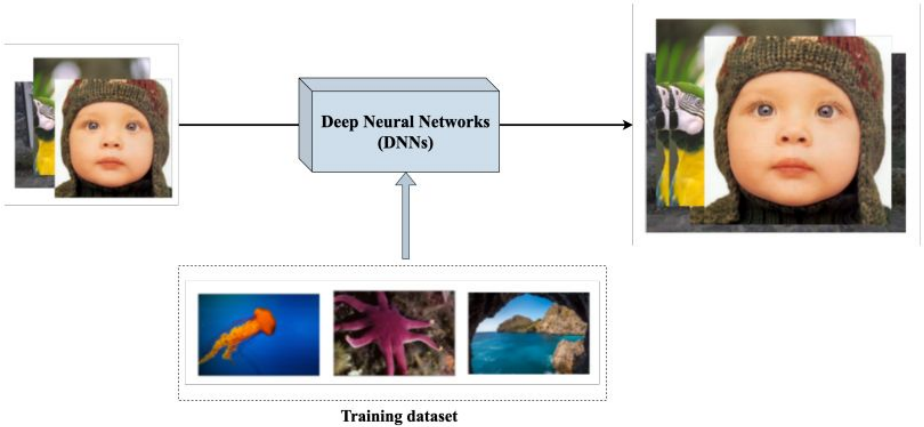
1. **The reconstruction-based methods.**

   Needs prior knowledge to define constraints for the target HR image. Typically, techniques like edge sharpening, regularization, and deconvolution are employed in this category.

2. **The learning-based methods.**

   These types of SISR methods use machine-learning techniques to estimate HR images. Pixel-based methods and example based methods are typical methods in this category. Other techniques like sparse coding and neighbor embedding are also widely used.

Deep Learning based SISR



Learning Strategy

- Supervised
- Semi-Supervised
- Unsupervised

Datasets

- Set4
- Set15
- BSD100
- DIV2K

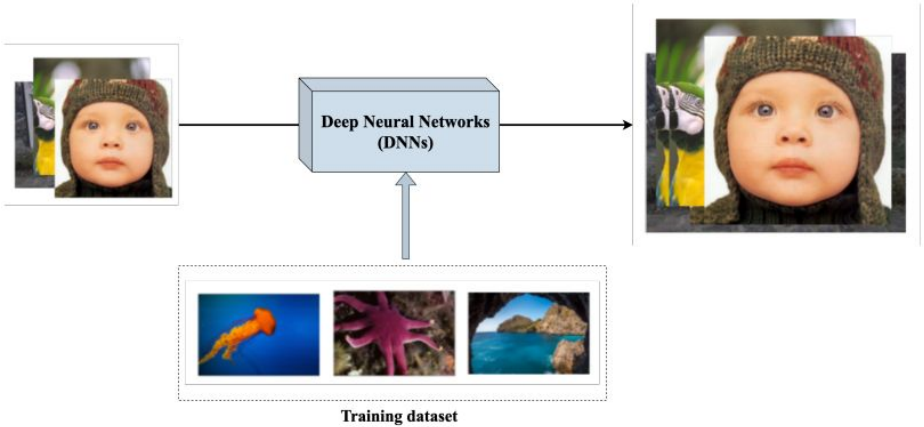
Loss Function

- Pixel loss
- Content loss
- Adversarial loss

Assessment Methods

- Full-reference metrics
  - PSNR
  - SSIM
- No-reference metrics
  - MOS
  - NIQE

Deep Learning based SISR



Learning Strategy

- **Supervised**
- **Semi-Supervised**
- Unsupervised

Datasets

- Set4
- Set15
- BSD100
- DIV2K

Loss Function

- Pixel loss
- Content loss
- Adversarial loss

Assessment Methods

- Full-reference metrics
  - PSNR
  - SSIM
- No-reference metrics
  - MOS
  - NIQE

Datasets

Datasets

“Easy” to be generated: every high resolution dataset can be degraded.

Two main elements needed:

1. The actual dataset:
   a. BSDS300[1]
   b. DIV2K[2]
   c. Flickr2K[3]

2. A degradation process:

$$\mathcal{D}(I_{HR};\theta_{\mathcal{D}}) = (I_{HR} \circledast \mathcal{K}) \downarrow_s + n$$

   a. degradation model as a combination of several operations
      i. BI: bicubic downsampling operation
      ii. BD: HR images are blurred by a Gaussian kernel, then downsampled with the scaling factor of 3.
      iii. DN: bicubic downsampling is performed on the HR image with scaling factor ×3, and then Gaussian noise is added.

[1] https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/
[2] https://data.vision.ee.ethz.ch/cvl/DIV2K/
[3] https://opendatalab.com/Flickr2K/download

Datasets

Name	Datasets	Short Description
Classical SR Training	T91	91 images for training
	BSDS200	A subset (train) of BSD500 for training
	General100	100 images for training
Classical SR Testing	set5	Set5 test dataset
	set14	Set14 test dataset
	BSDS100	A subset (test) of BSD500 for testing
	urban100	100 building images for testing (regular structures)
	manga109	109 images of Japanese manga for testing
	historical	10 gray LR images without the ground-truth
2K Resolution	DIV2K	proposed in NTIRE17 (800 train and 100 validation)
	Flickr2K	2650 2K images from Flickr for training
	DF2K	A merged training dataset of DIV2K and Flickr2K
OST (Outdoor Scenes)	OST Training	7 categories images with rich textures
	OST300	300 test images of outdoor scences
PIRM	PIRM	PIRM self-val, val, test datasets

Assessment Methods

Metrics for image assessment in SISR - Full-reference

Common metrics for image quality evaluation.

- **Peak to Signal Noise Ratio – PSNR:**
  - Ratio between signal and noise

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE(\hat{Y},Y)} \right) \qquad MSE(\hat{Y}, Y) = \frac{\sum_i (\hat{y}_i - y_i)^2}{N}$$

- **Structural Similarity Index – SSIM:**
  - Index of perceptual quality of the image

$$SSIM(X,Y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Metrics for image assessment in SISR - No-reference

- **Mean opinion score - MOS**: a number of viewers rate their opinions on the quality of a set of images by Double-stimulus (i.e. every viewer has both the source and test images).
- **Natural Image Quality Evaluator - NIQE**: a completely blind image quality assessment method. It extracts a set of local (quality-aware) features from images based on a natural scene statistic (NSS) model, then fits the feature vectors to a multivariate Gaussian (MVG) model.

$$\mathcal{D}(v_1, v_2, \Sigma_1, \Sigma_2) = \sqrt{((v_1 - v_2)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} (v_1 - v_2))}$$

- **Ma**: learning-based (regression forests) no-reference image quality assessment. It learns from perceptual scores based on human subject studies involving a large number of SR images.
- **Perception index - PI**: is first proposed to evaluate the perceptual quality. It is a combination of the no-reference image quality measures Ma and NIQE:

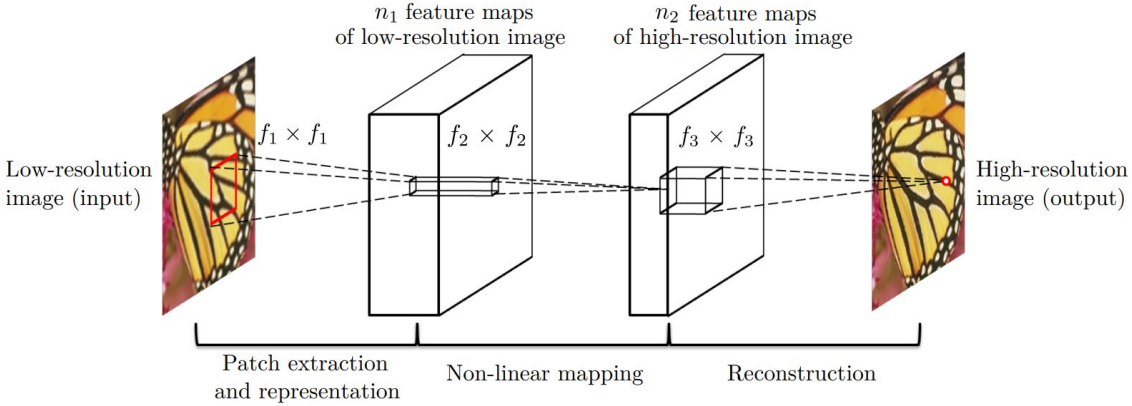
$$PI = \frac{1}{2}((10 - Ma) + NIQE)$$

State Of The Art

SRCNN [2015]

Image Super-Resolution Using Deep Convolutional Networks

Pre-upscaling : the image is first upscaled with a bicubic upsampling, then is processed by 3 convolutional layers in order to improve the details in the image.



Mean Squared Error - MSE / L2 Loss function:

$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n} ||\Phi(\uparrow I_{LR}, \theta) - I_{Target}||^2$$

$\Phi(\uparrow I_{LR}, \theta)$

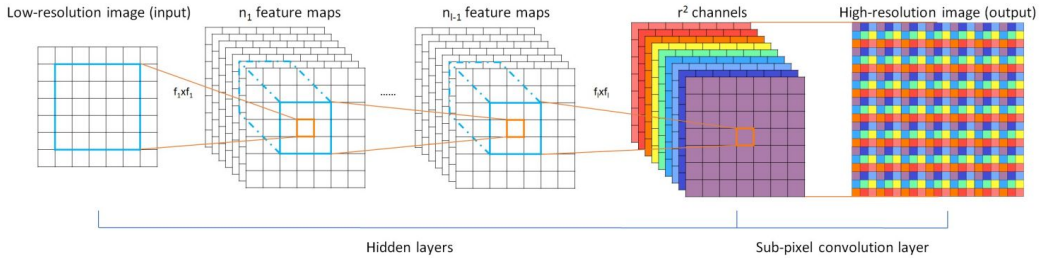
I_{Target}

ESPCN [2016]

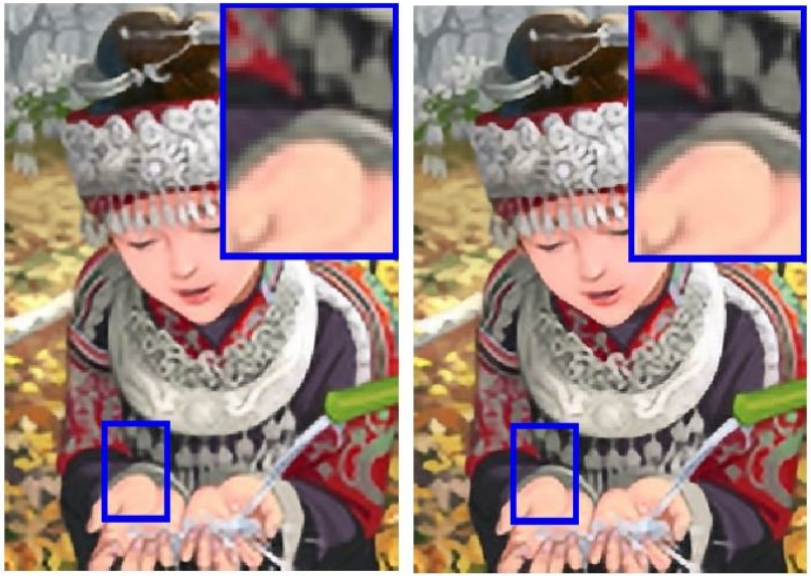
Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network

Post feature processing upsampling:

2 conv layers for features extraction + **Pixel Shuffle** layer



Works on low resolution image, means much lighter approach!



SRResNet & SRGAN [2016]

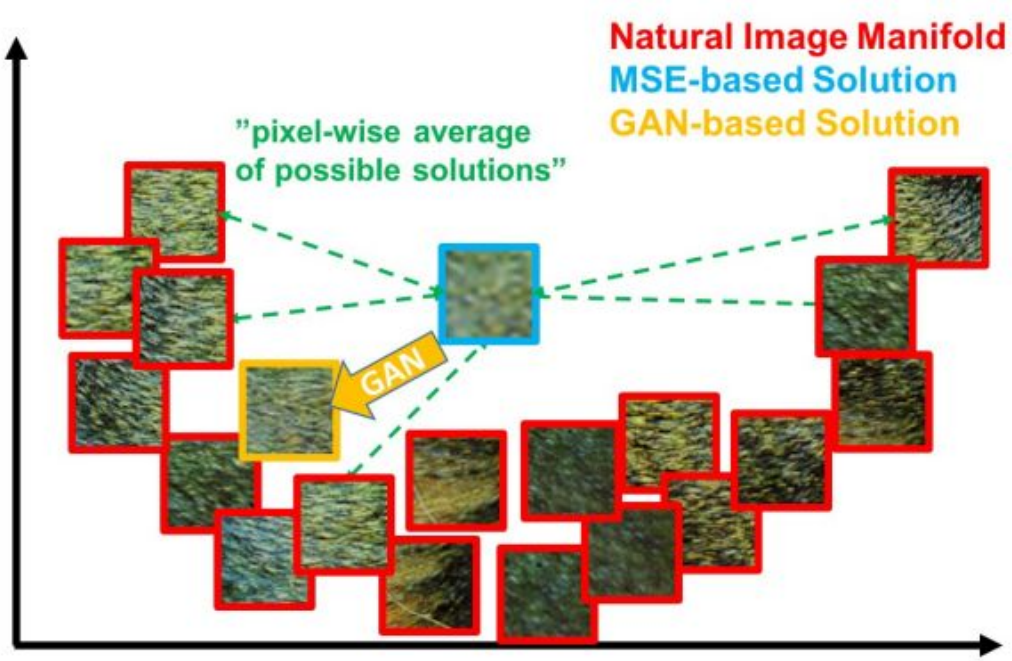
Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

2 proposed solutions:

- Deep ResNet style model
- Generative Adversarial Networks (GAN) framework adopted

Extra contribution:

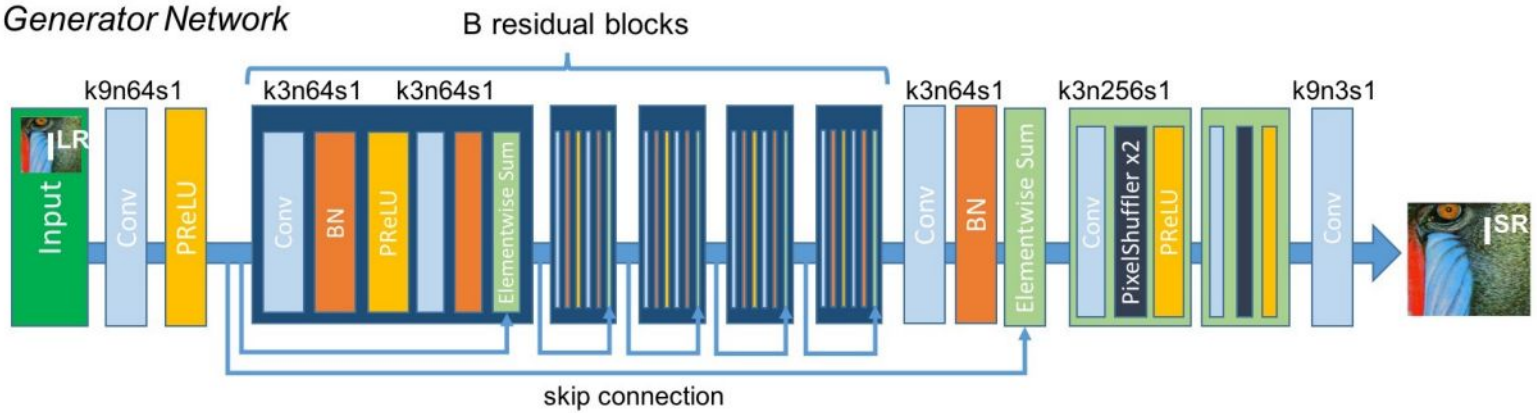
- Perceptual quality in loss function

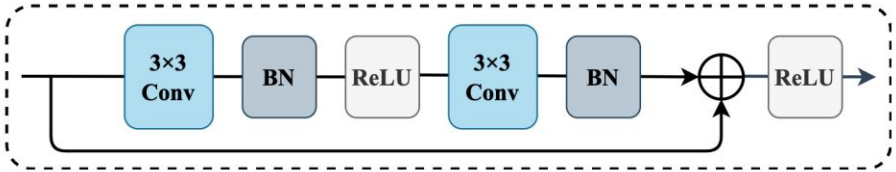


SRResNet & SRGAN [2016]

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

SRResNet: very deep model that adopts resnets skip connections to avoid vanishing gradients.





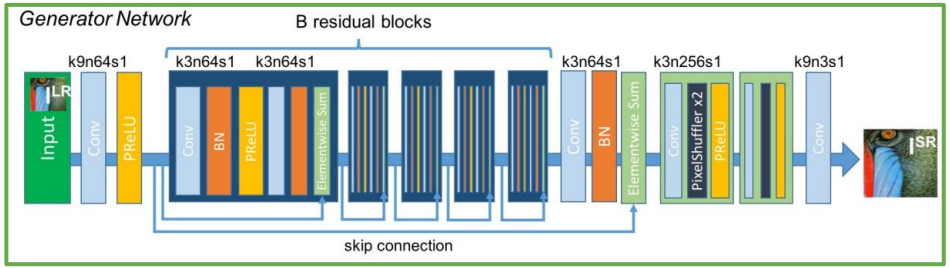
Trained using L2 distance

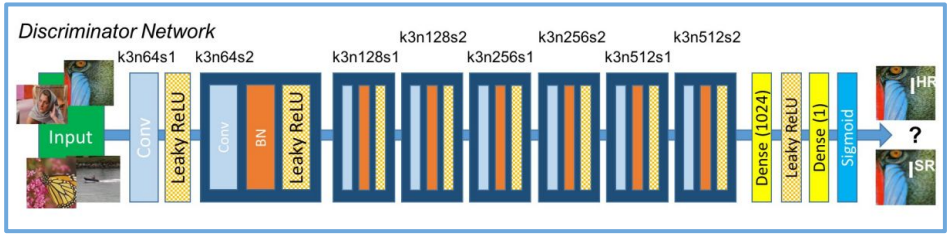
$$L_{MSE} = \frac{1}{n}\sum_{i=1}^{n} ||\Phi(\uparrow I_{LR}, \theta) - I_{Target}||^2$$

SRResNet & SRGAN [2016]

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

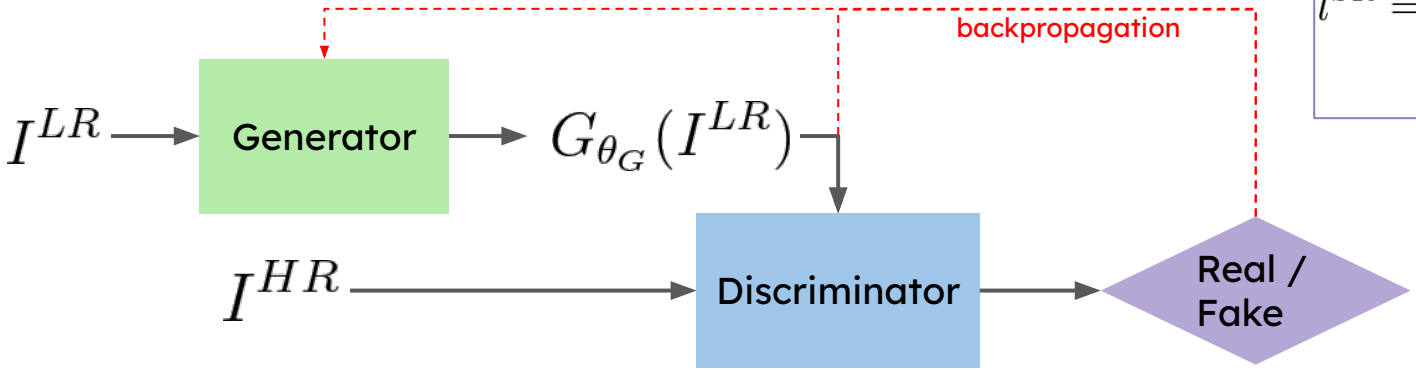
SRGAN: Generative/Discriminative networks trained with weighted complex loss function.





Perceptual loss function

$$l^{SR} = \underbrace{l_X^{SR}}_{\text{content loss}} + \underbrace{10^{-3} l_{gen}^{SR}}_{\text{adversarial loss}}$$



SRResNet & SRGAN [2016]

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

Perceptual loss function

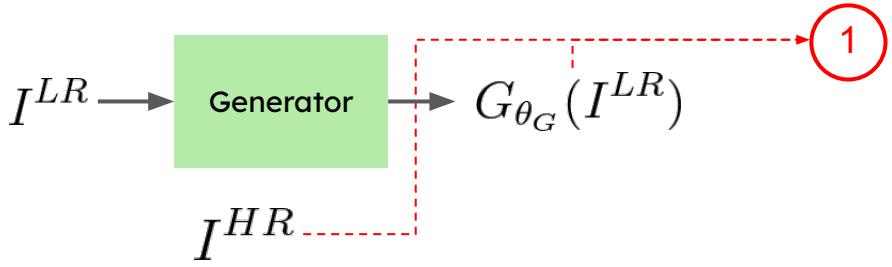
$$l^{SR} = \underbrace{l_X^{SR}}_{\text{content loss}} + \underbrace{10^{-3} l_{gen}^{SR}}_{\text{adversarial loss}}$$

SRResNet & SRGAN [2016]

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

Perceptual loss function

$$l^{SR} = \underbrace{l_X^{SR}}_{\text{content loss}} + \underbrace{10^{-3} l_{gen}^{SR}}_{\text{adversarial loss}}$$



1 Content Loss:

MSE / Per pixel Loss:

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

Perceptual Loss:

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

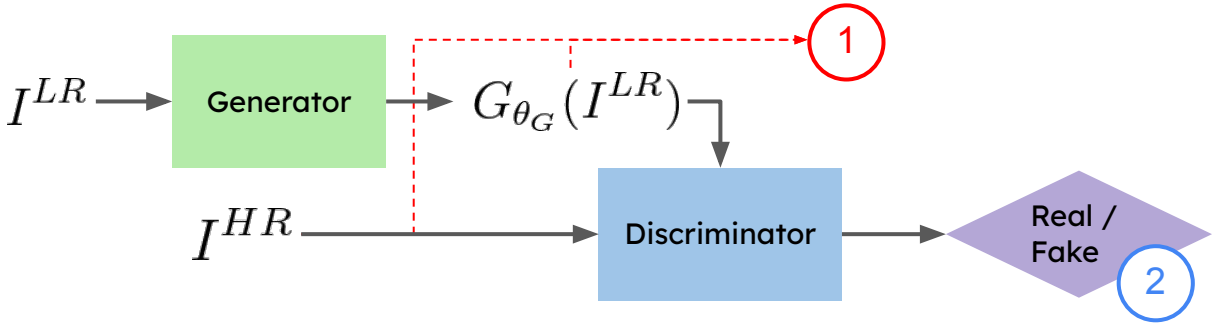
$\phi_{i,j}$: feature map obtained by the j-th convolution (after activation) before the i-th maxpooling layer within the VGG19 network

SRResNet & SRGAN [2016]

Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

Perceptual loss function

$$l^{SR} = \underbrace{l_X^{SR}}_{\text{content loss}} + \underbrace{10^{-3} l_{gen}^{SR}}_{\text{adversarial loss}}$$



1 Content Loss:

MSE / Per pixel Loss:

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

Perceptual Loss:

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

$\phi_{i,j}$: feature map obtained by the j-th convolution (after activation) before the i-th maxpooling layer within the VGG19 network

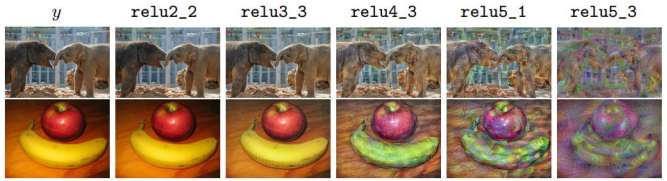
2 Adversarial Loss:

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

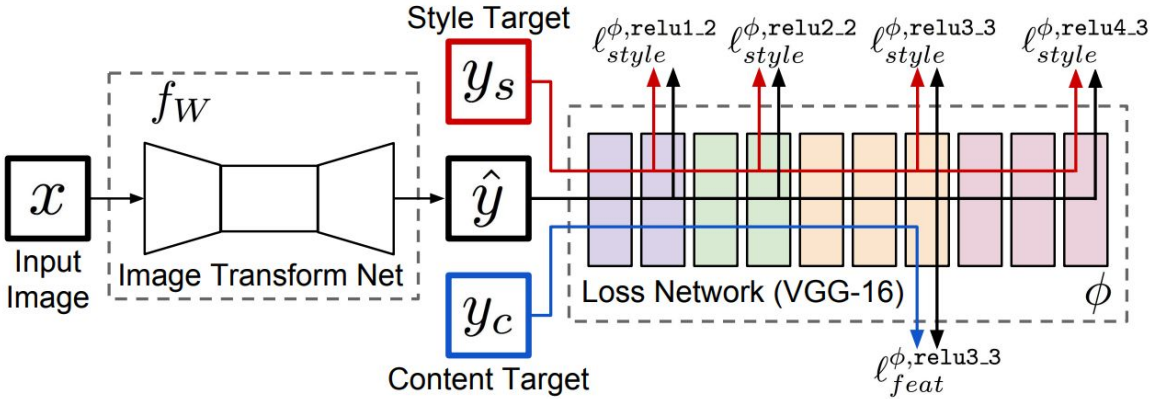
From the definition of Adversarial training framework

Perceptual Loss

Perceptual (VGG) Loss - original version



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



Analysis of images at features level.

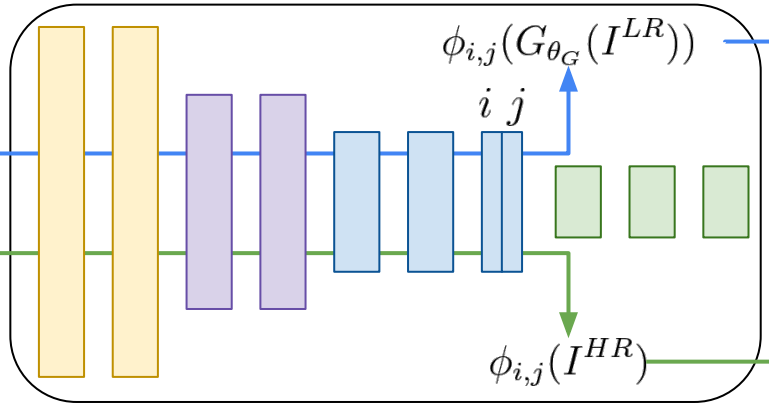
Originally used for style transfer, in SISR we only use the content features at a specific level of depth.

Perceptual (VGG) Loss - SISR version

$G_{\theta_G}(I^{LR})$

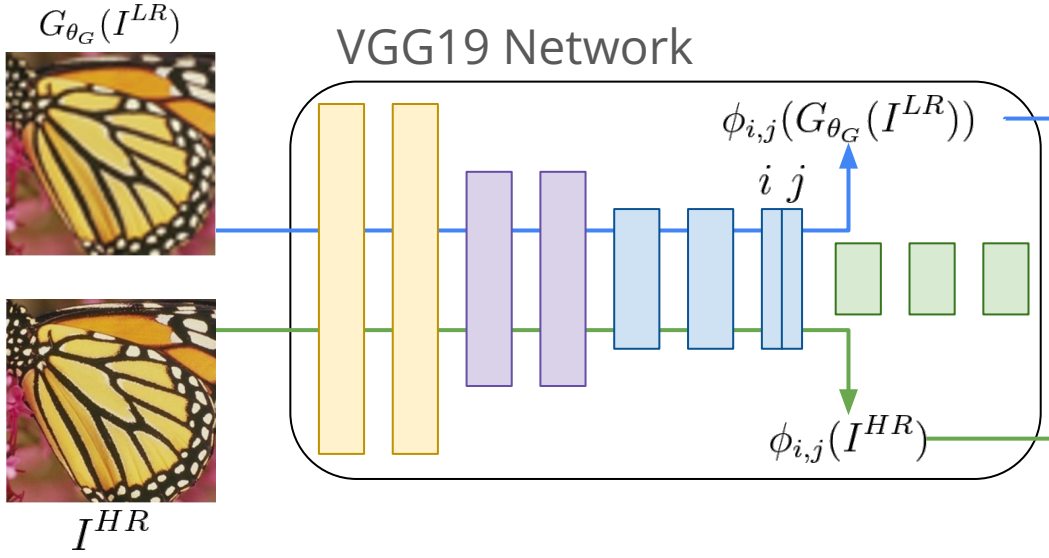
I^{HR}

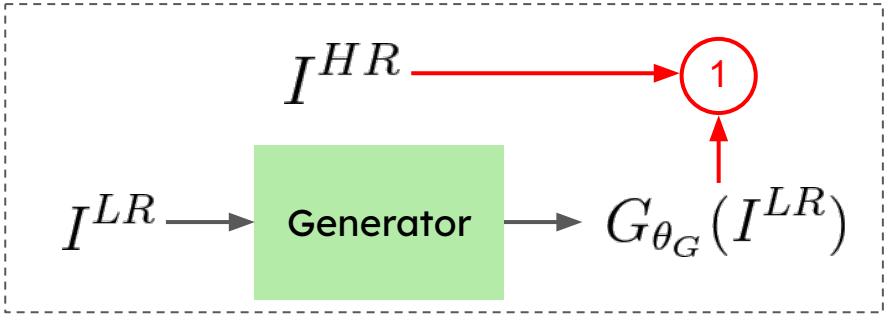
VGG19 Network



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

Perceptual (VGG) Loss - SISR version





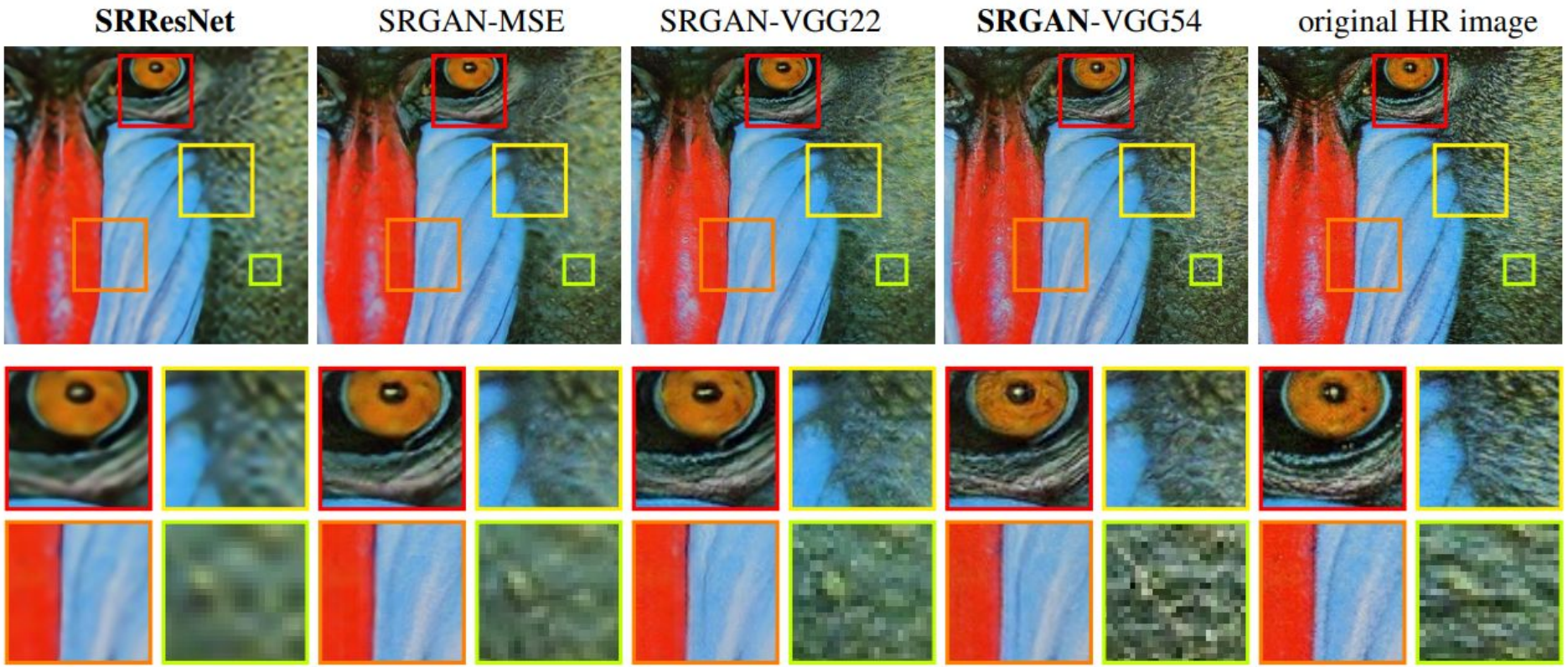
$$l^{SR}_{VGG/i,j} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2$$

+

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

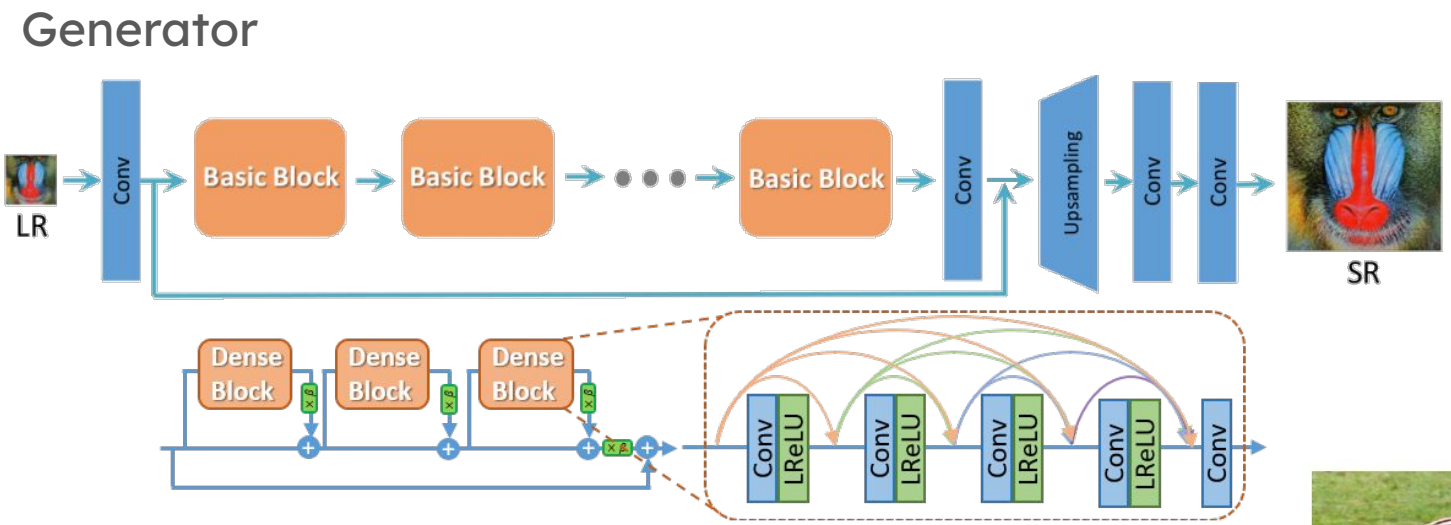
SRResNet & SRGAN [2016]

Results



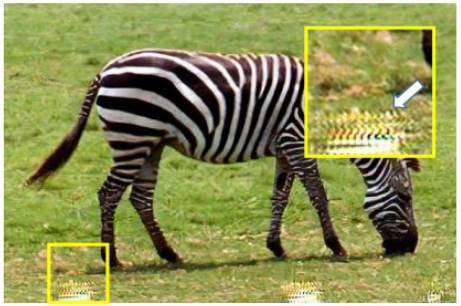
ESRGAN [2018]

ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks

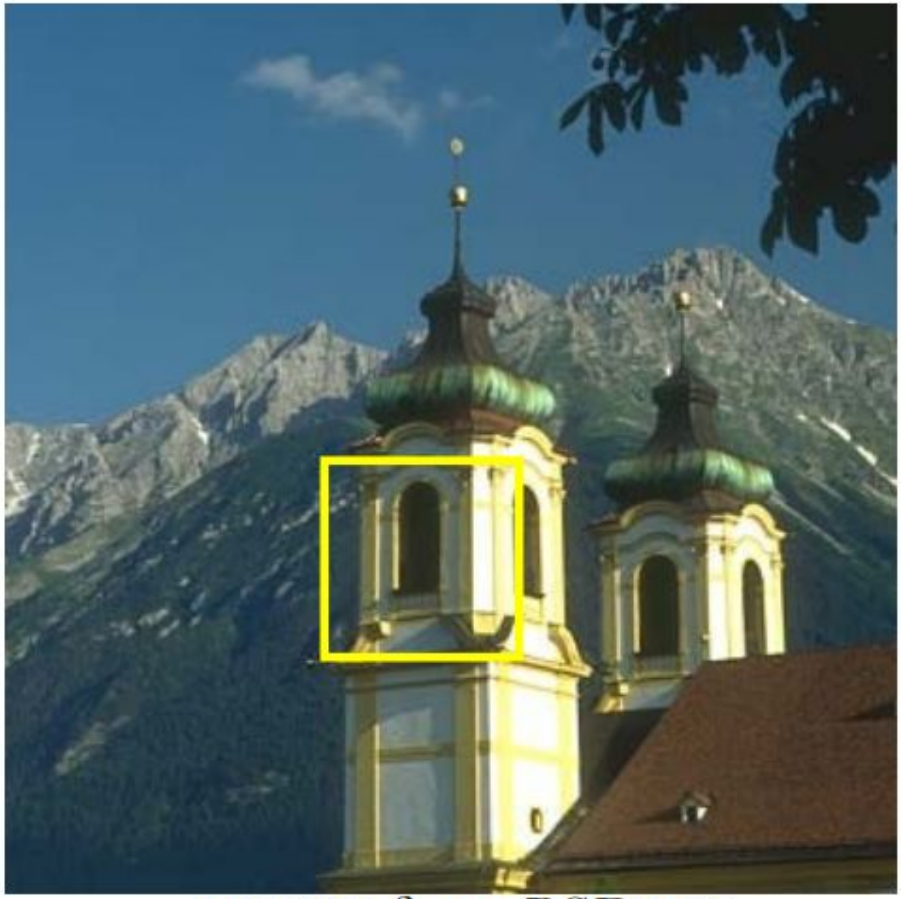


Evolution of SRGAN:

- Even deeper architecture
- Analysis on the impact of elements of the architecture

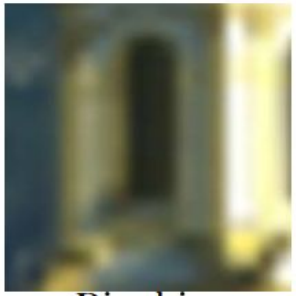


zebra from Set14

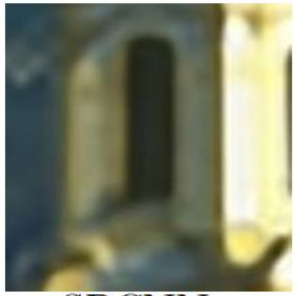


126007 from BSD100
(PSNR / Percpetual Index)

HR
(∞ / 2.16)



Bicubic
(27.25 / 6.61)



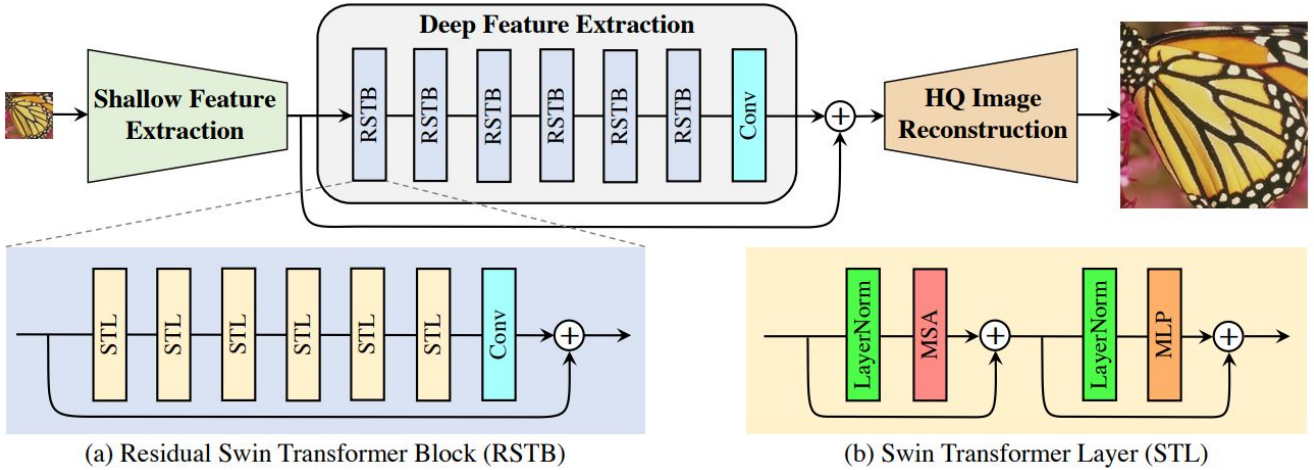
SRCNN
(28.40 / 5.70)

SRGAN
(26.31 / 2.46)

ESRGAN(ours)
(26.52 / 2.42)

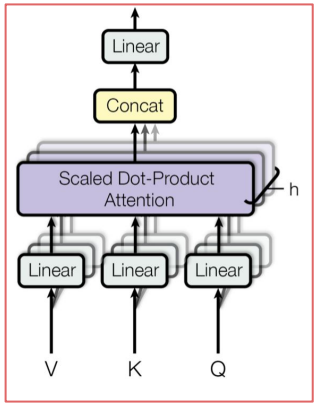
SwinIR [2021]

SwinIR: Image Restoration Using Swin Transformer



(a) Residual Swin Transformer Block (RSTB)

(b) Swin Transformer Layer (STL)



Multi-head self-attention Module (MSA)

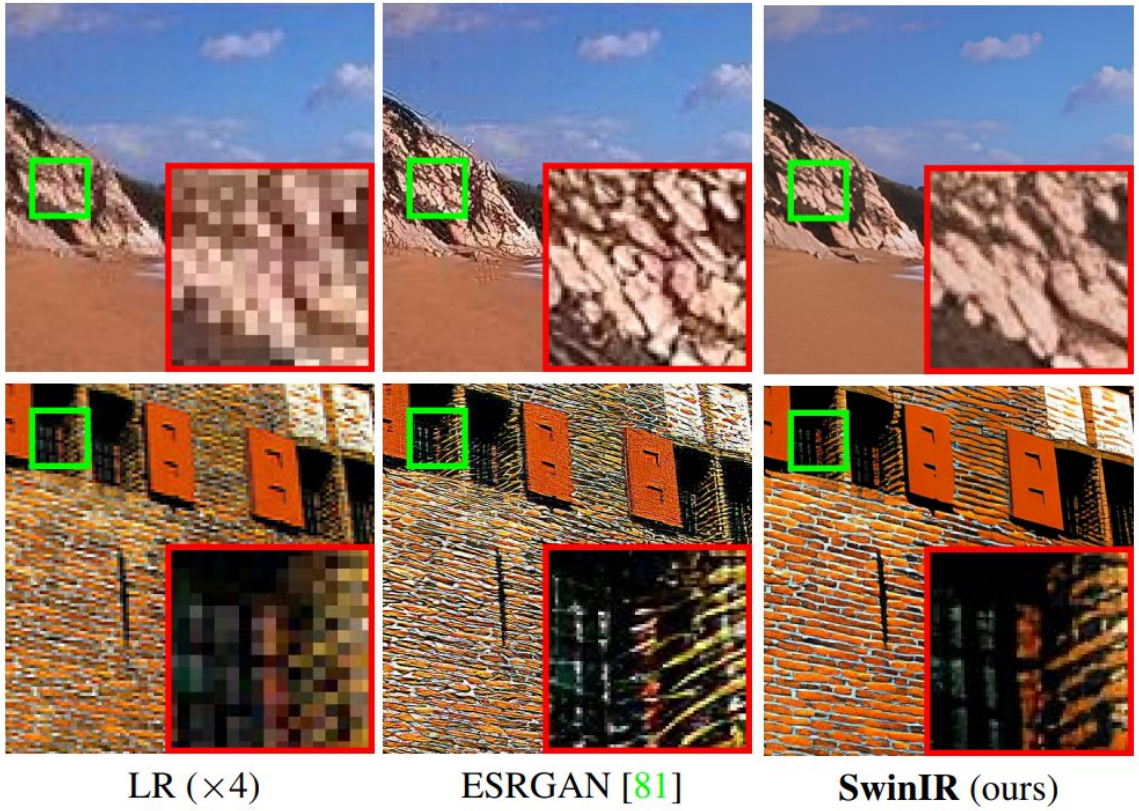
Adaptation of Swin Transformers to Image Restoration tasks.

VERY Deep Model (36 Swin Transformers feature extractor layers)

Real-SR task Loss: L1 + GAN + Perceptual Loss

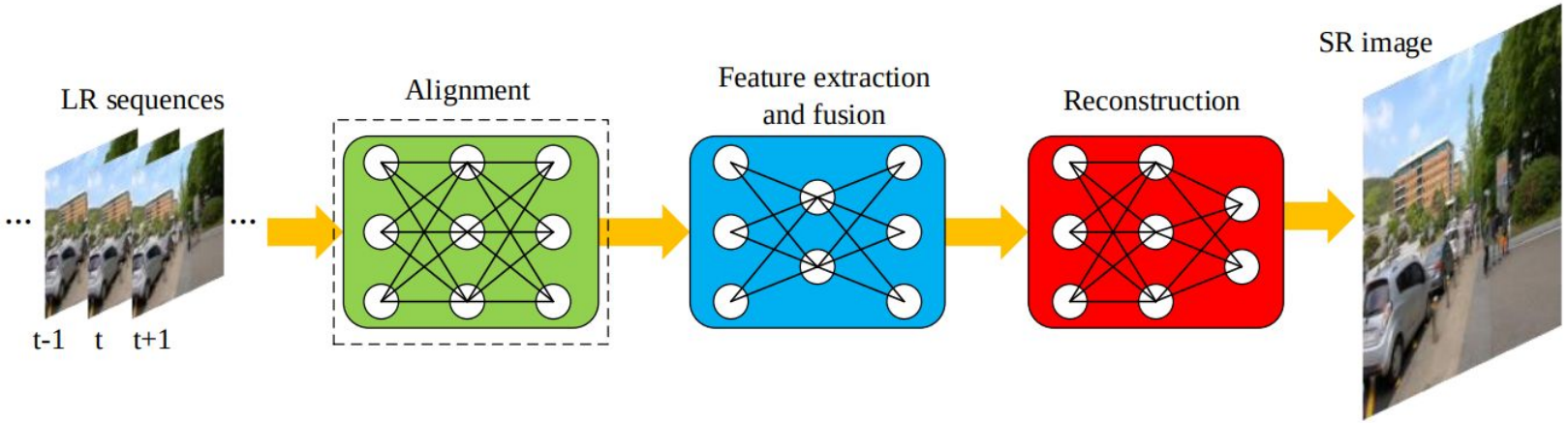
SwinIR [2021]

SwinIR: Image Restoration Using Swin Transformer



What’s next?

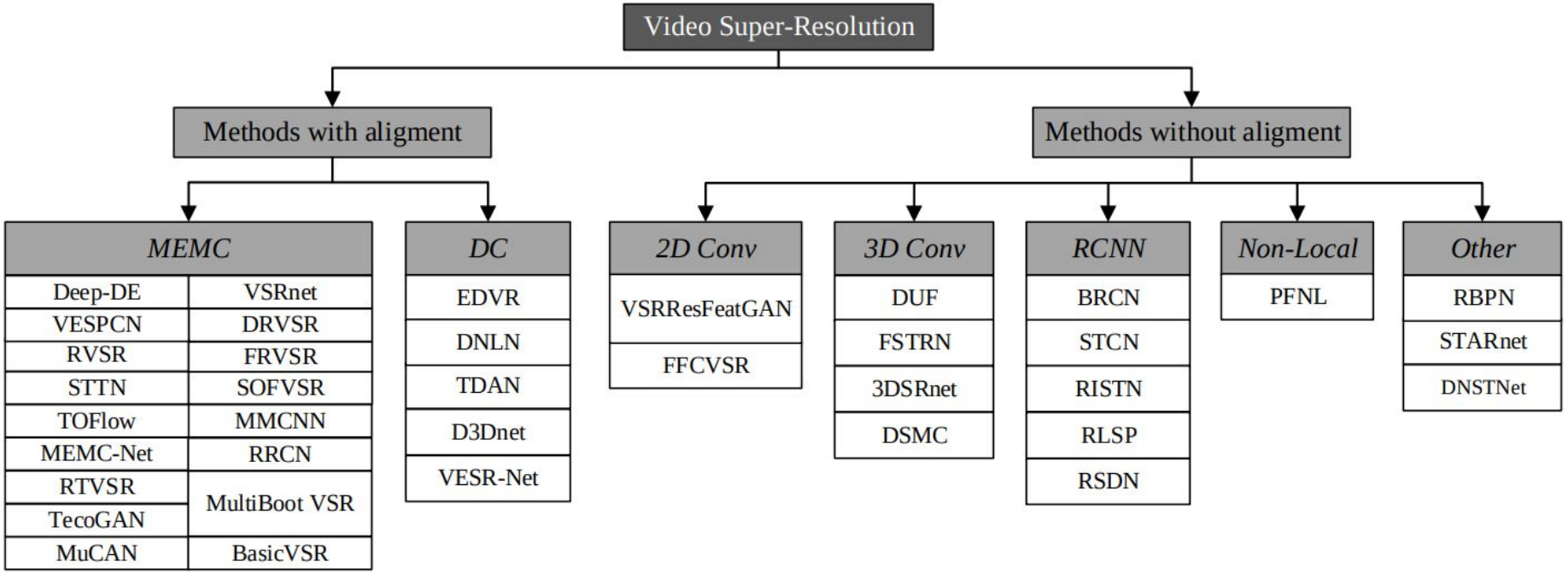
Video Super Resolution



Enhancement of video sequences

- Multiple input frames
  - Alleviates the ill nature of the problem
  - More challenges! e.g. frames needs alignment, object occlusion from one frame to the next one etc…
  - How to effectively leverage the information from neighboring frames
- More data means increase computational and memory costs

Video Super Resolution



Very active research field!