清華大學
Tsinghua University

商汤
sensetime

Chapter 2 - Section 10

Low-Level Computer Vision Task

Dr. Zou Dongqing & Dr. Zhang Yu

Thursday, April 23, 2021

Outline

Part 1 Image Restoration: Problems and Methods

Part 2 Generative Image Models for Image Restoration

商汤 sensetime

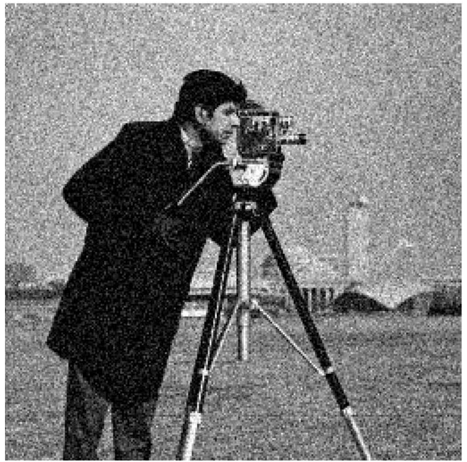
清華大學 Tsinghua University

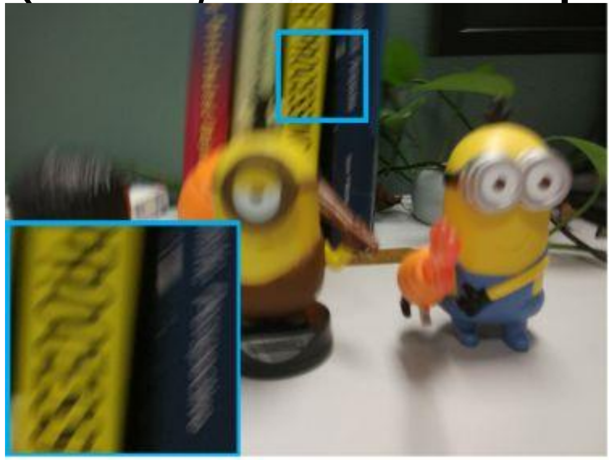
Image Restoration

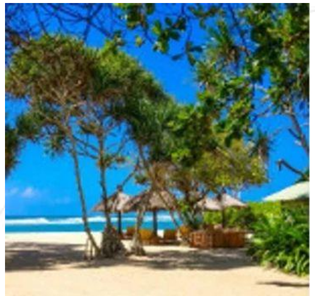
Part 1	Image Restoration: Problems and Applications
Part 2	Sparse Representation for Image Restoration
Part 3	Low Rank Minimization for Image Restoration
Part 4	Deep Learning for Image Restoration

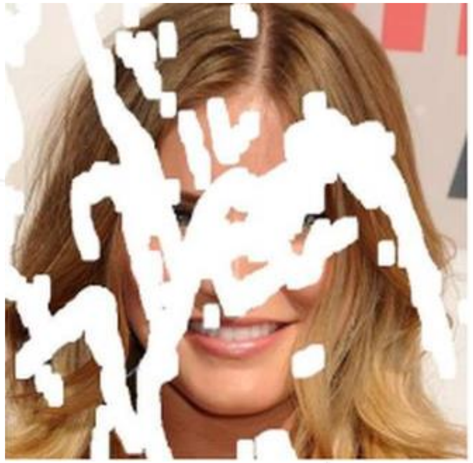
Image restoration: the problem and applications

- Reconstruct the latent image(clear, high-resolution, …) from its degraded measurement (noise, down-sampling, …)

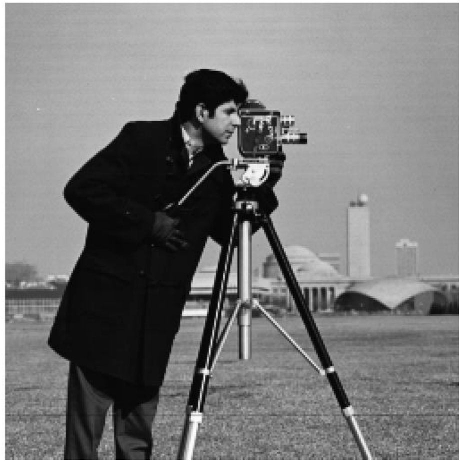

Noisy

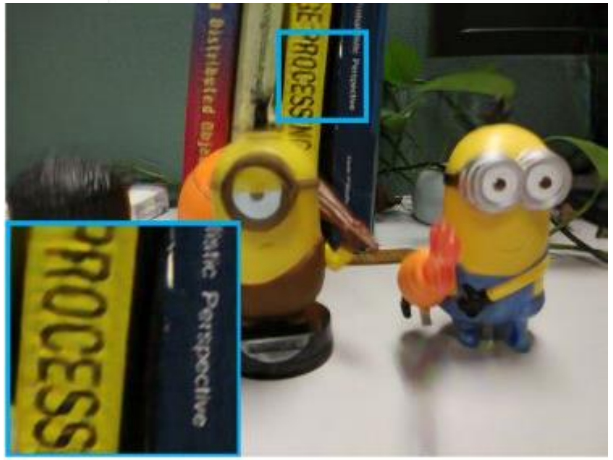

Blur

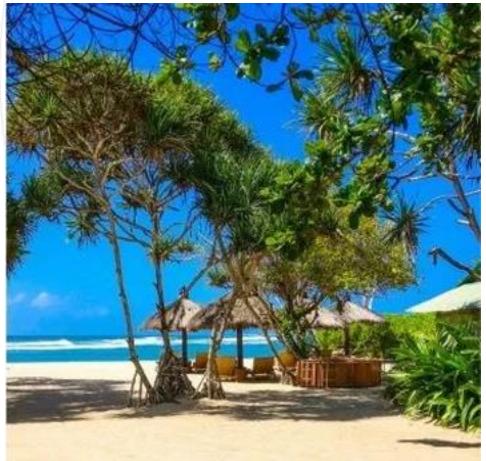

Down-sampling


Damaged

……


Denoising


Deblurring


Super-Resolution

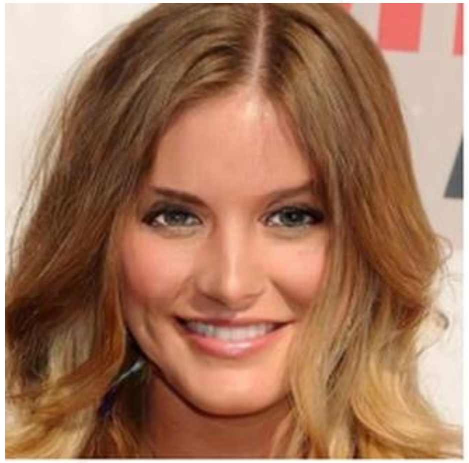

Inpainting

Image restoration: the problem and applications

- Reconstruct the latent image (clear, high-resolution, …) from its observed image or degraded measurement (noise, down-sampling, …)

- General observation model

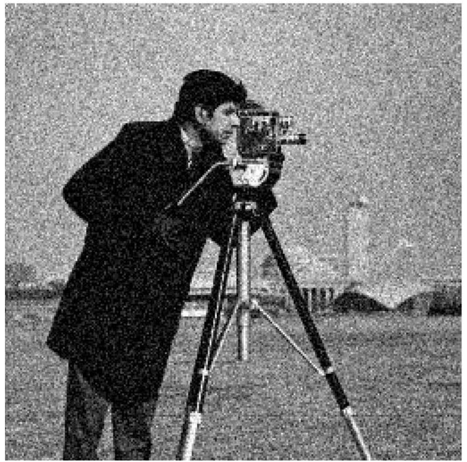
$$y = Hx + n$$

- H： the observation (degradation) matrix
- n： the additive noise

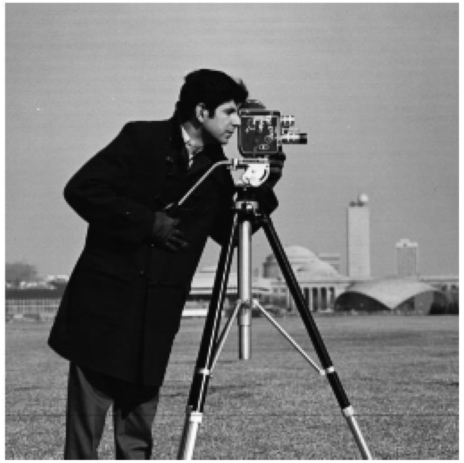
- Image restoration is a typical *ill-posed* inverse problem. *Prior information* is needed to solve it.

Image restoration: the problem and applications

- Reconstruct the latent image(clear, high-resolution, ...) from its degraded measurement (noise, down-sampling, ...)



Noisy



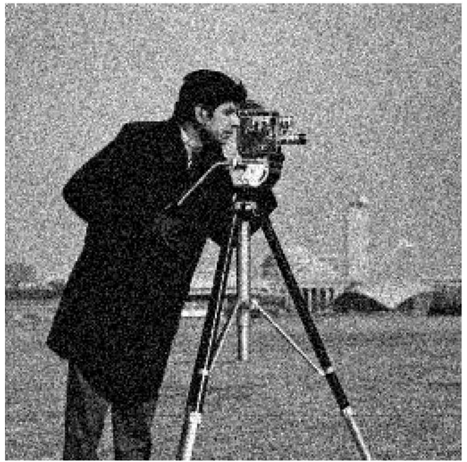
Denoising

H is an identity matrix

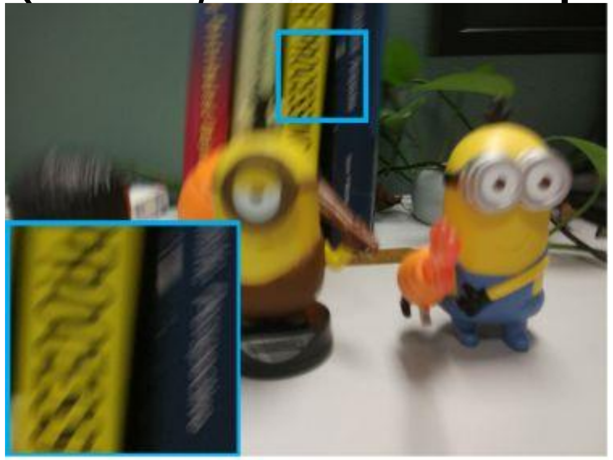
N is a random noise matrix

Image restoration: the problem and applications

- Reconstruct the latent image(clear, high-resolution, …) from its degraded measurement (noise, down-sampling, …)

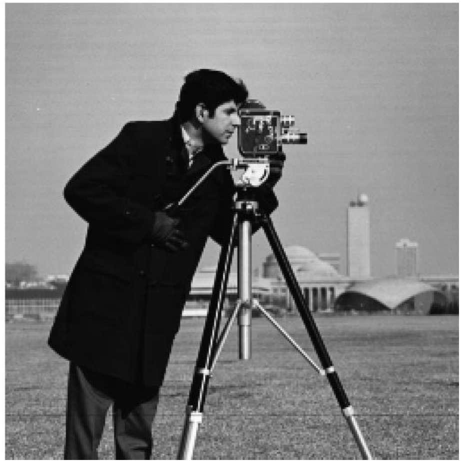


Noisy

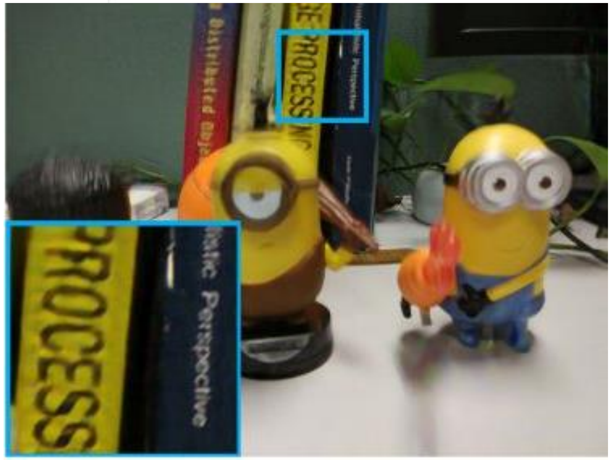


Blur

H is a blurring matrix



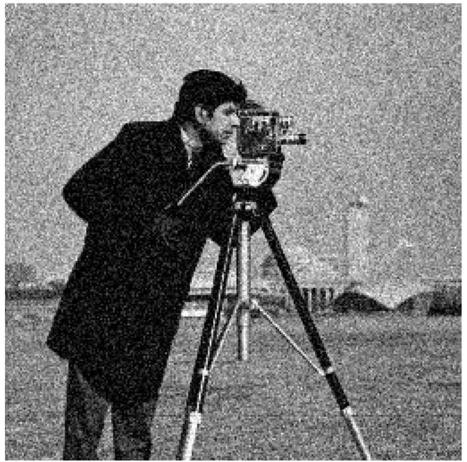
Denoising



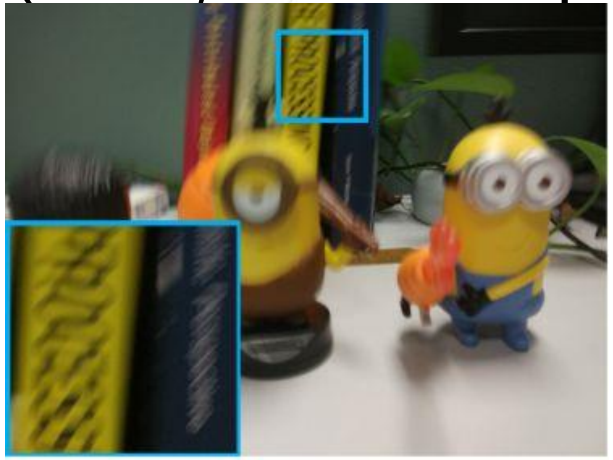
Deblurring

Image restoration: the problem and applications

- Reconstruct the latent image(clear, high-resolution, …) from its degraded measurement (noise, down-sampling, …)



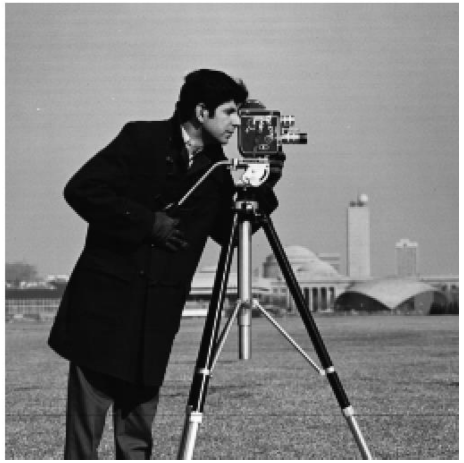
Noisy



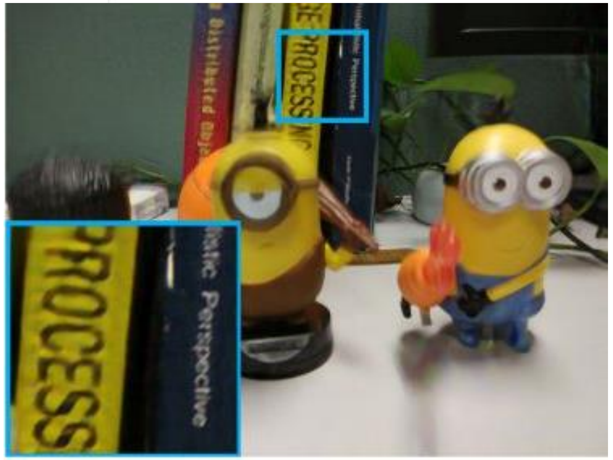
Blur

Down-sampling

H is a compound matrix of blurring and down-sampling



Denoising

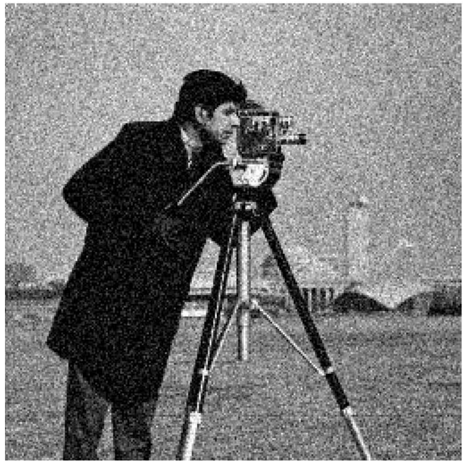


Deblurring

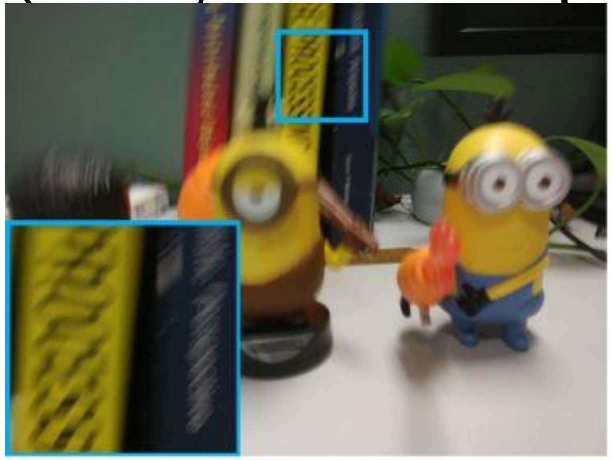
Super-Resolution

Image restoration: the problem and applications

- Reconstruct the latent image(clear, high-resolution, …) from its degraded measurement (noise, down-sampling, …)

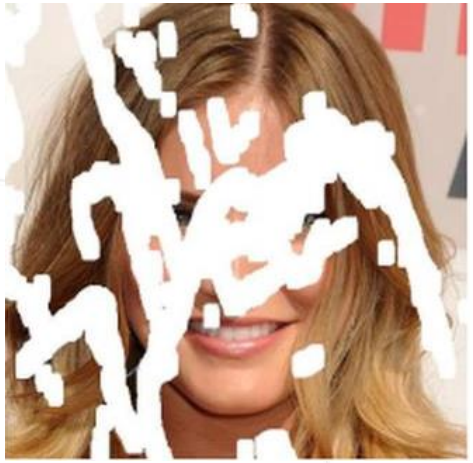


Noisy



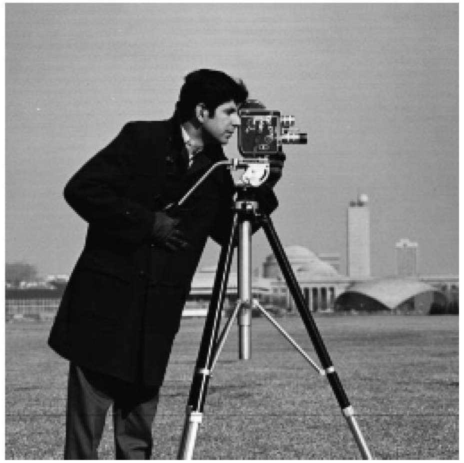
Blur

Down-sampling

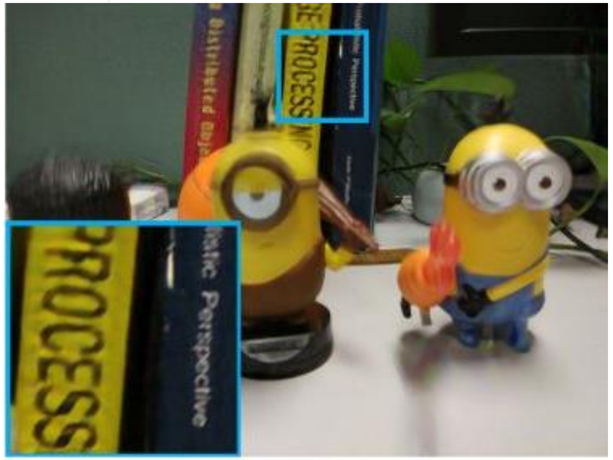


Damaged

H is a mask of damaged pixels

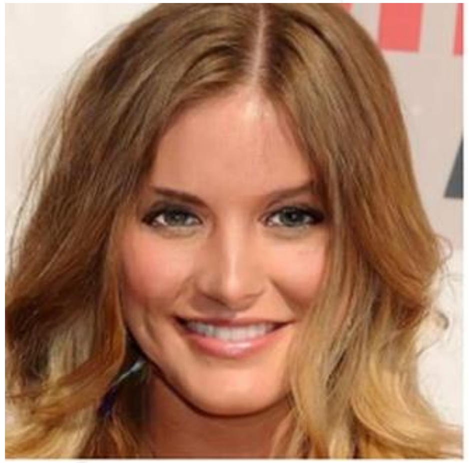


Denoising



Deblurring

Super-Resolution



Inpainting

Image restoration: the problem and applications

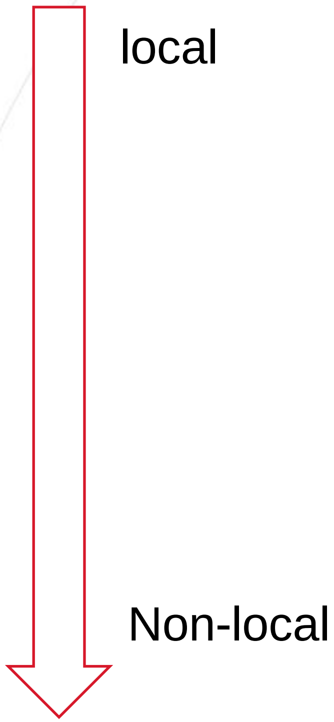
- Methodology overview: from classic computer vision to deep learning

- Filtering based methods
- Transform based methods
- Model based optimization
- Deep learning

Image restoration: the problem and applications

- Methodology overview: from classic computer vision to deep learning

- Filtering based methods
- Gaussian filtering
- PDE-based anisotropic diffusion
- Bilateral filtering
- Nonlocal means filtering
- Domain transfer based filtering
- ……



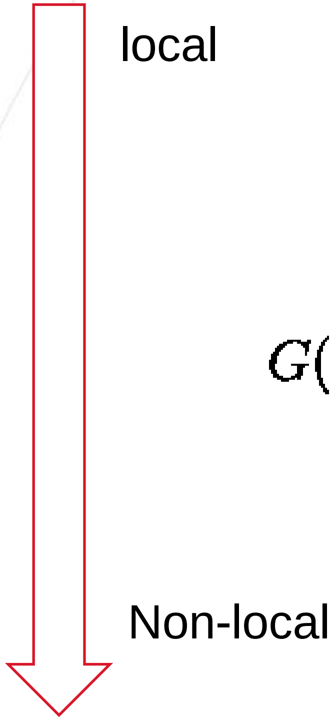
local

Non-local

Image restoration: the problem and applications

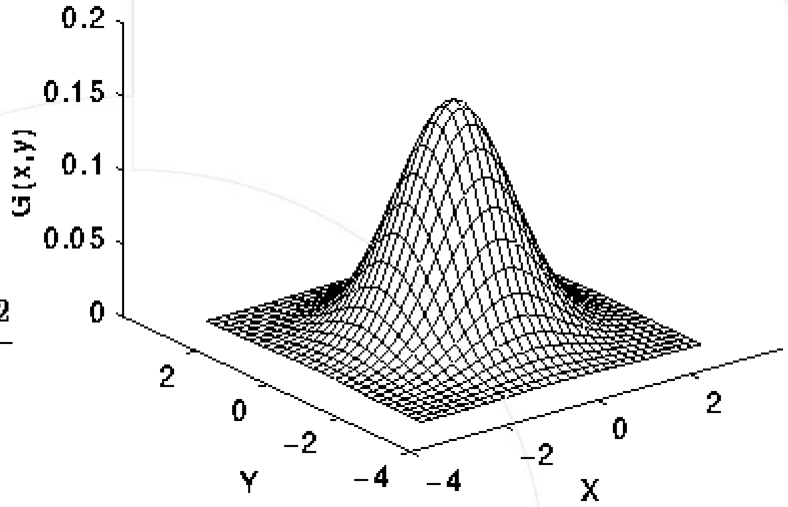
- Methodology overview: from classic computer vision to deep learning

- Filtering based methods
- Gaussian filtering
- PDE-based anisotropic diffusion
- Bilateral filtering
- Nonlocal means filtering
- Domain transfer based filtering
- ……


local

Non-local

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

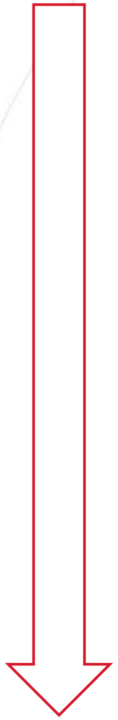


Smoothing edges while removing noise

Image restoration: the problem and applications

- Methodology overview: from classic computer vision to deep learning

- Filtering based methods
- Gaussian filtering
- PDE-based anisotropic diffusion
- Bilateral filtering
- Nonlocal means filtering
- Domain transfer based filtering
- ……

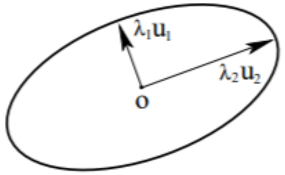


local

Non-local

A diffusion tensor D is assimilated to a 2 × 2 symmetric and positive-definite matrix, having then two positive eigenvalues λ1, λ2 and two associated orthonormal eigenvectors u1⊥u2.

$$\mathbf{D} = \begin{pmatrix} a & b \\ b & c \end{pmatrix} = \lambda_1 \, \mathbf{u}_1\mathbf{u}_1^T + \lambda_2 \, \mathbf{u}_2\mathbf{u}_2^T$$

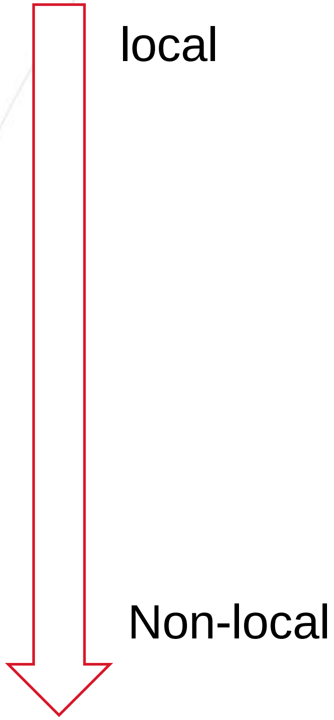


Preserving better edges than low-pass filtering

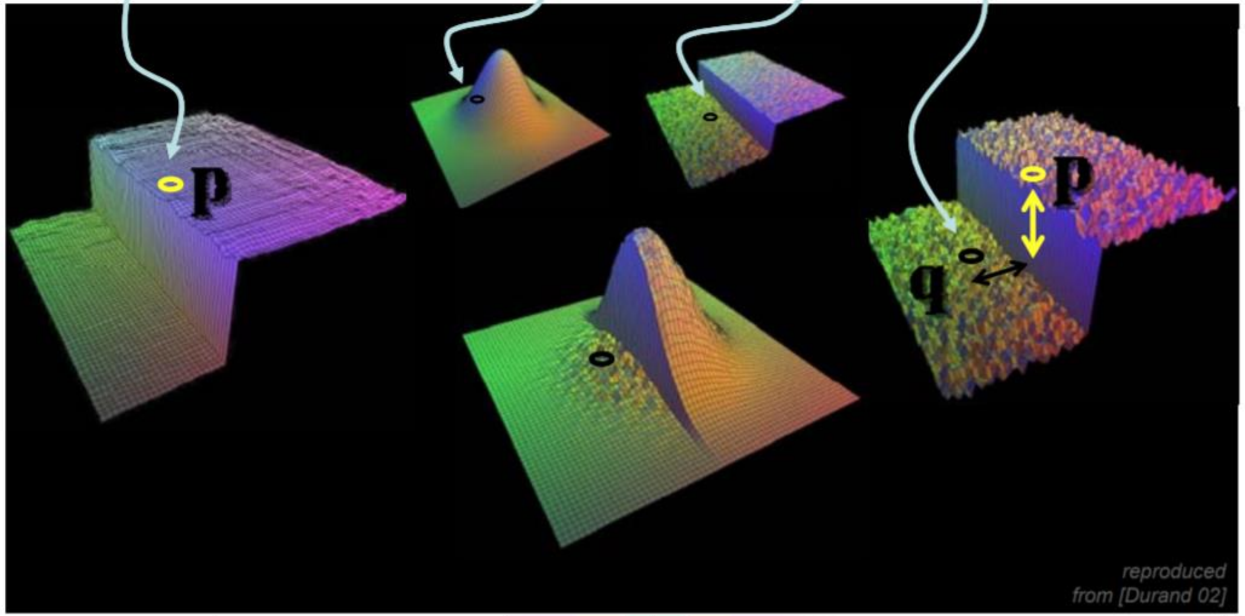
Image restoration: the problem and applications

- Methodology overview: from classic computer vision to deep learning

- Filtering based methods
- Gaussian filtering
- PDE-based anisotropic diffusion
- Bilateral filtering
- Nonlocal means filtering
- Domain transfer based filtering
- ……



$$\underbrace{BF[I]_{\mathbf{p}}}\ = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q}\in S} \underbrace{G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)}\ \underbrace{G_{\sigma_r}(|I_{\mathbf{p}}-I_{\mathbf{q}}|)}\ I_{\mathbf{q}}$$



Exploiting both spatial and intensity similarity

- 1D image = line of pixels
- Better visualized as a plot

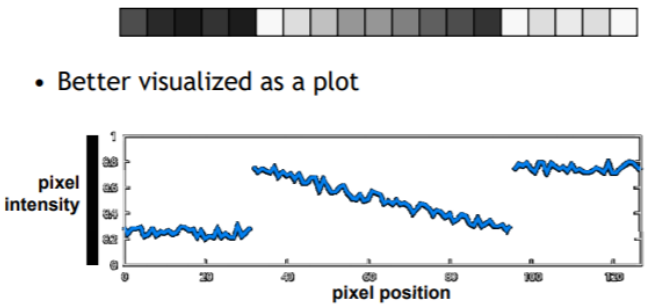
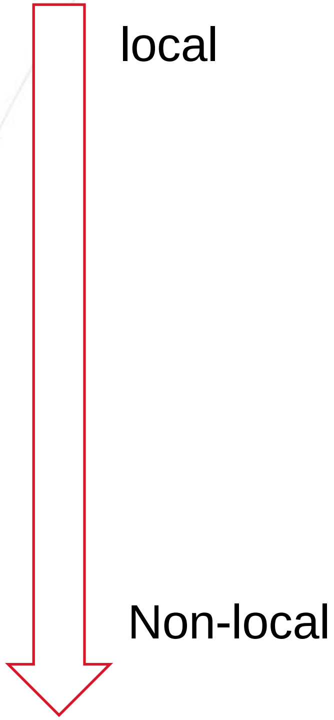
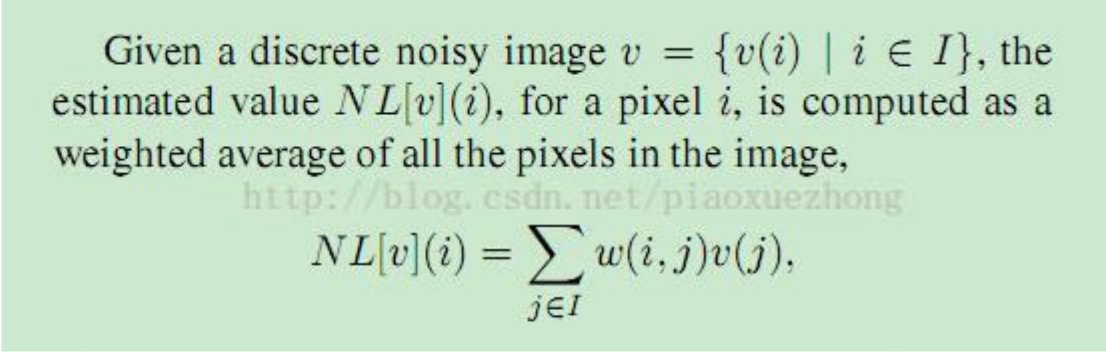


Image restoration: the problem and applications

- Methodology overview: from classic computer vision to deep learning

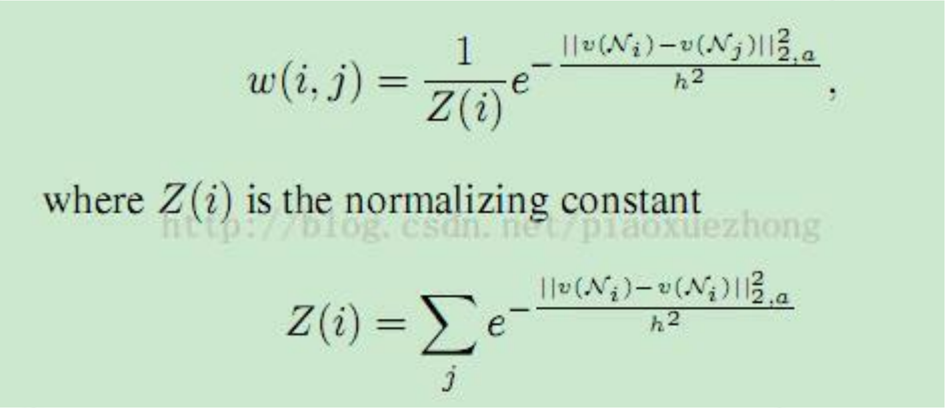
- Filtering based methods
- Gaussian filtering
- PDE-based anisotropic diffusion
- Bilateral filtering
- Nonlocal means filtering
- Domain transfer based filtering
- ……





Given a discrete noisy image $v = \{v(i) \mid i \in I\}$, the estimated value $NL[v](i)$, for a pixel i, is computed as a weighted average of all the pixels in the image,

$$NL[v](i) = \sum_{j \in I} w(i,j)v(j),$$



$$w(i,j) = \frac{1}{Z(i)} e^{-\frac{\|v(\mathcal{N}_i)-v(\mathcal{N}_j)\|^2_{2,a}}{h^2}},$$

where $Z(i)$ is the normalizing constant

$$Z(i) = \sum_j e^{-\frac{\|v(\mathcal{N}_i)-v(\mathcal{N}_i)\|^2_{2,a}}{h^2}}$$

Exploiting the nonlocal self similarity

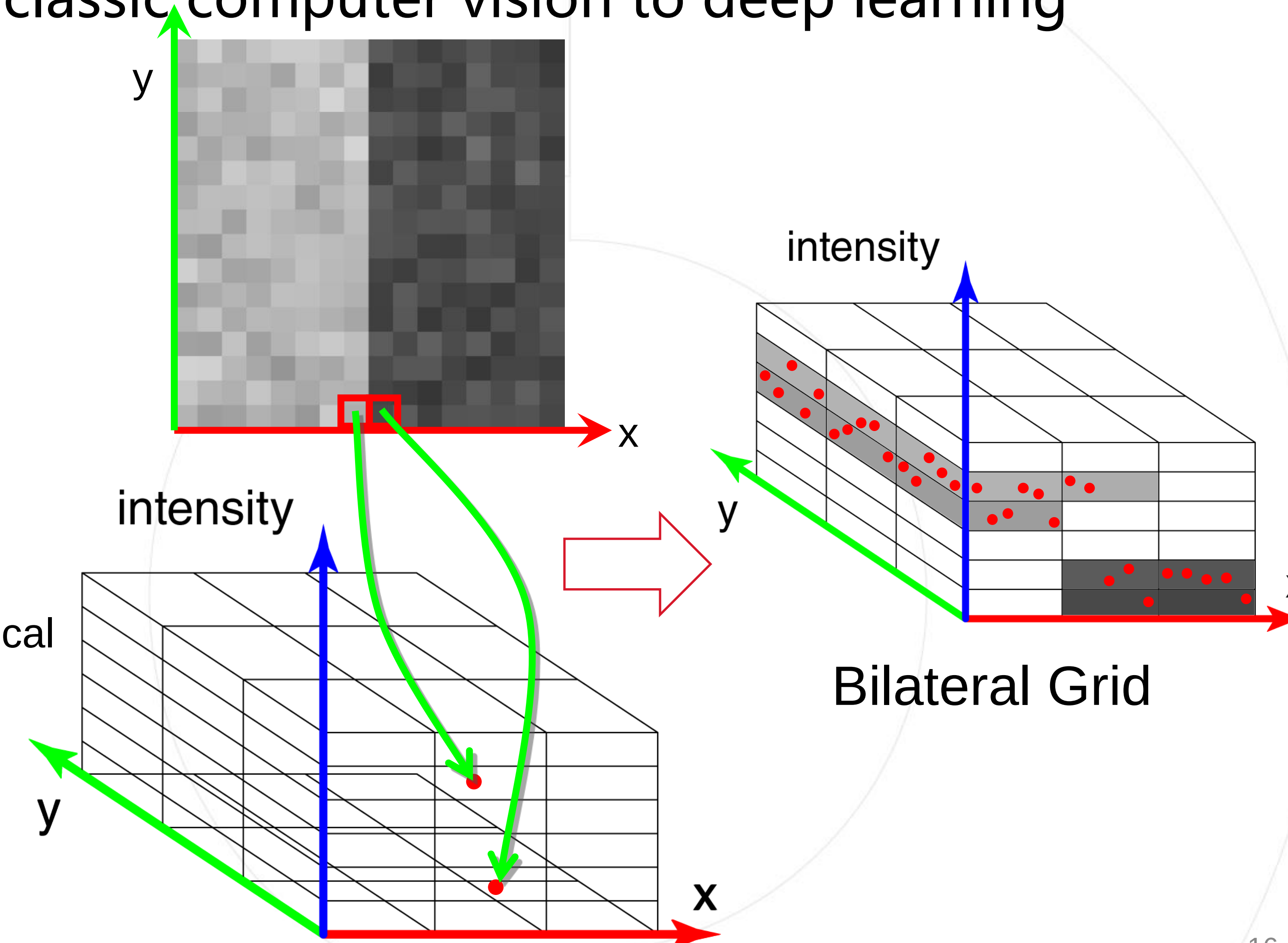
Image restoration: the problem and applications

- Methodology overview: from classic computer vision to deep learning

- Filtering based methods
- Gaussian filtering
- PDE-based anisotropic diffusion
- Bilateral filtering
- Nonlocal means filtering
- Domain transfer based filtering
- ……

local

Non-local



Bilateral Grid

The bilateral grid, that enables fast edge-aware image processing

Image restoration: the problem and applications

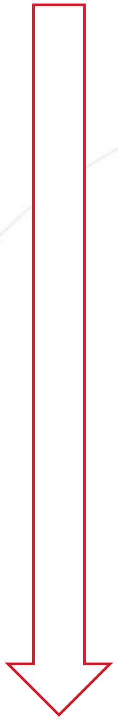
- Methodology overview: from classic computer vision to deep learning

- Filtering based methods

- Gaussian filtering
- PDE-based anisotropic diffusion
- Bilateral filtering
- Nonlocal means filtering
- Domain transfer based filtering
- ……

- Transform based methods

- Fourier transform
- Wavelet transform
- Curvelet transform
- Ridgelet transform
- Bandlet transform
- ……



global bases

Local and Multiscale Geometric bases

Image restoration: the problem and applications

- Methodology overview: from classic computer vision to deep learning

- Filtering based methods
- Gaussian filtering
- PDE-based anisotropic diffusion
- Bilateral filtering
- Nonlocal means filtering
- Domain transfer based filtering
- ……

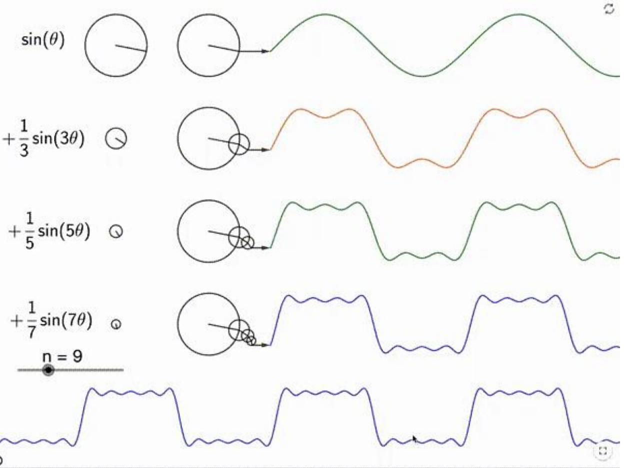
- Transform based methods
- Fourier transform
- Wavelet transform
- Curvelet transform
- Ridgelet transform
- Bandlet transform
- ……

Virtually everything in the world can be described via a waveform - a function of time, space or some other variable.

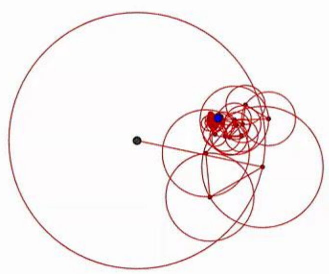
All waveforms, no matter what you scribble or observe in the universe, are actually just the sum of simple sinusoids of different frequencies.

$$s_N(x) \overset{?}{=} \frac{a_0}{2} + \sum_{n=1}^{N} \left(\overbrace{a_n}^{A_n \sin(\phi_n)} \cos(2\pi f n x) + \overbrace{b_n}^{A_n \cos(\phi_n)} \sin(2\pi f n x) \right)$$

$$X(w) = \sum_{i=0}^{N-1} x(t_i) e^{-jwt_i}$$



$\sin(\theta)$
$+\frac{1}{3}\sin(3\theta)$
$+\frac{1}{5}\sin(5\theta)$
$+\frac{1}{7}\sin(7\theta)$
n = 9



“big” sine and cosine wave bases

Image restoration: the problem and applications

- Methodology overview: from classic computer vision to deep learning

- Filtering based methods
- Transform based methods

- Gaussian filtering
- PDE-based anisotropic diffusion
- Bilateral filtering
- Nonlocal means filtering
- Domain transfer based filtering
- ……

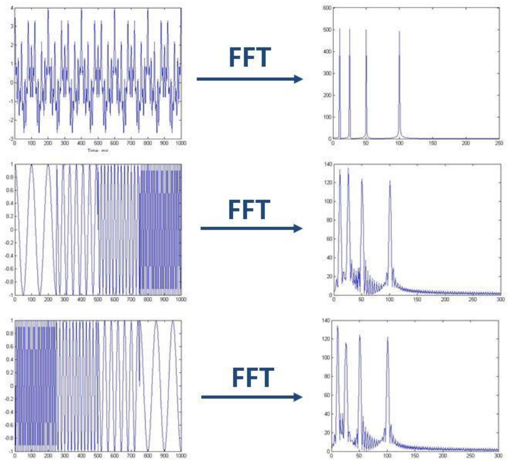
- Fourier transform
- Wavelet transform
- Curvelet transform
- Ridgelet transform
- Bandlet transform
- ……

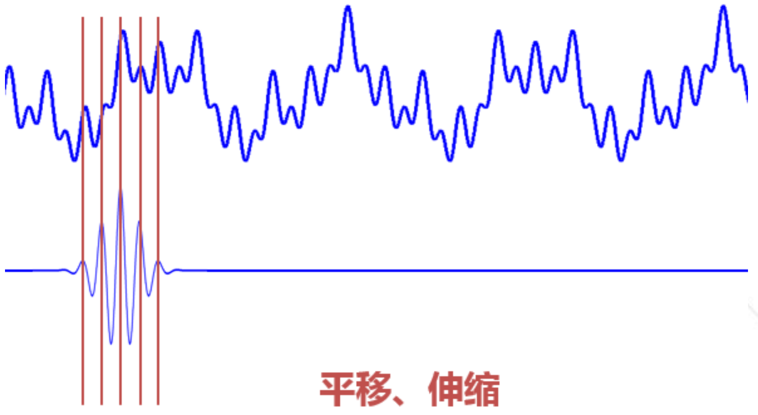
A major disadvantage of the Fourier Transform is it captures *global* frequency information

Wavelet Transform, which **decomposes a function into a set of wavelets**.

Wavelets have two basic properties: scale and location.

$$F(w) = \int_{-\infty}^{\infty} f(t) * e^{-iwt} dt \Rightarrow WT(a,\tau) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(t) * \psi\left(\frac{t-\tau}{a}\right) dt$$





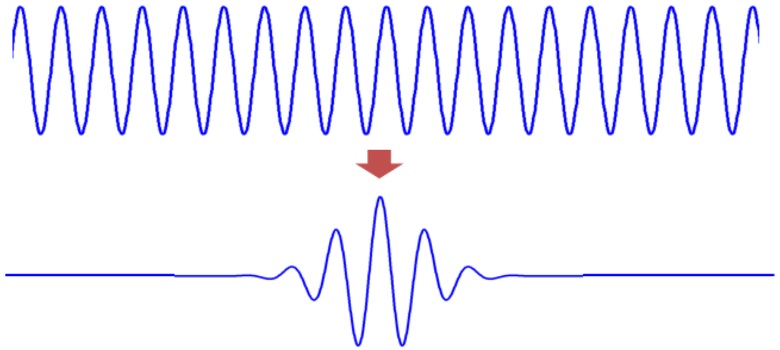


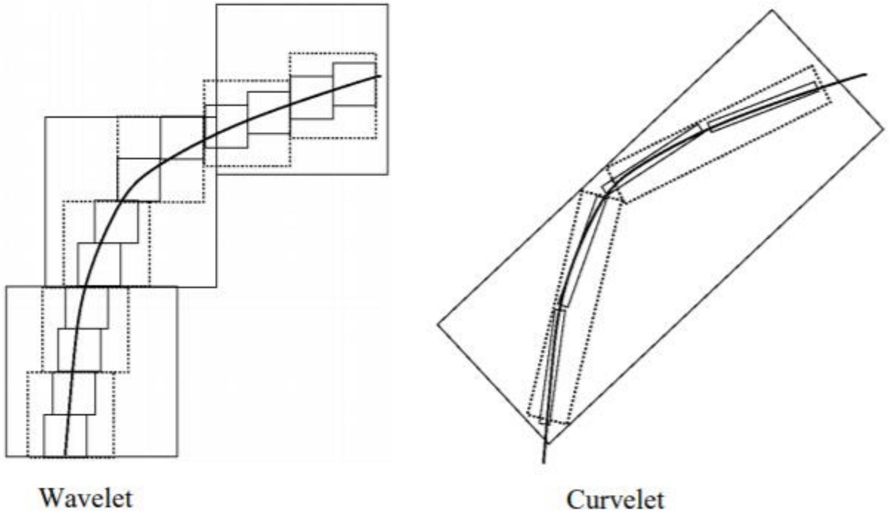
Image restoration: the problem and applications

- Methodology overview: from classic computer vision to deep learning

- Filtering based methods
  - Gaussian filtering
  - PDE-based anisotropic diffusion
  - Bilateral filtering
  - Nonlocal means filtering
  - Domain transfer based filtering
  - ……

- Transform based methods
  - Fourier transform
  - Wavelet transform
  - Curvelet transform
  - Ridgelet transform
  - Bandlet transform
  - ……

WT has only a fixed number of directional elements independent of scales.



From Fourier dictionary to curvelet dictionary and so on, the dictionary becomes more and more redundant and over-complete.

Image restoration: the problem and applications

- Methodology overview: from classic computer vision to deep learning

- Filtering based methods

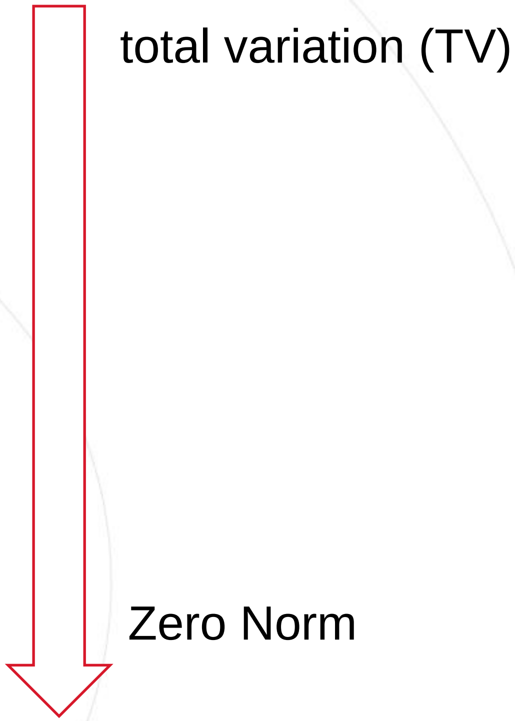
- Gaussian filtering
- PDE-based anisotropic diffusion
- Bilateral filtering
- Nonlocal means filtering
- Domain transfer based filtering
- ……

- Transform based methods

- Fourier transform
- Wavelet transform
- Curvelet transform
- Ridgelet transform
- Bandlet transform
- ……

- Model based optimization

- Regularization based
- Sparse representation
- Low-rank minimization
- ……



total variation (TV)

Zero Norm

Image restoration: the problem and applications

- Methodology overview: from classic computer vision to deep learning

- Filtering based methods
- Gaussian filtering
- PDE-based anisotropic diffusion
- Bilateral filtering
- Nonlocal means filtering
- Domain transfer based filtering
- ……

- Transform based methods
- Fourier transform
- Wavelet transform
- Curvelet transform
- Ridgelet transform
- Bandlet transform
- ……

- Model based optimization
- Regularization based
- Sparse representation
- Low-rank minimization
- ……

General model:

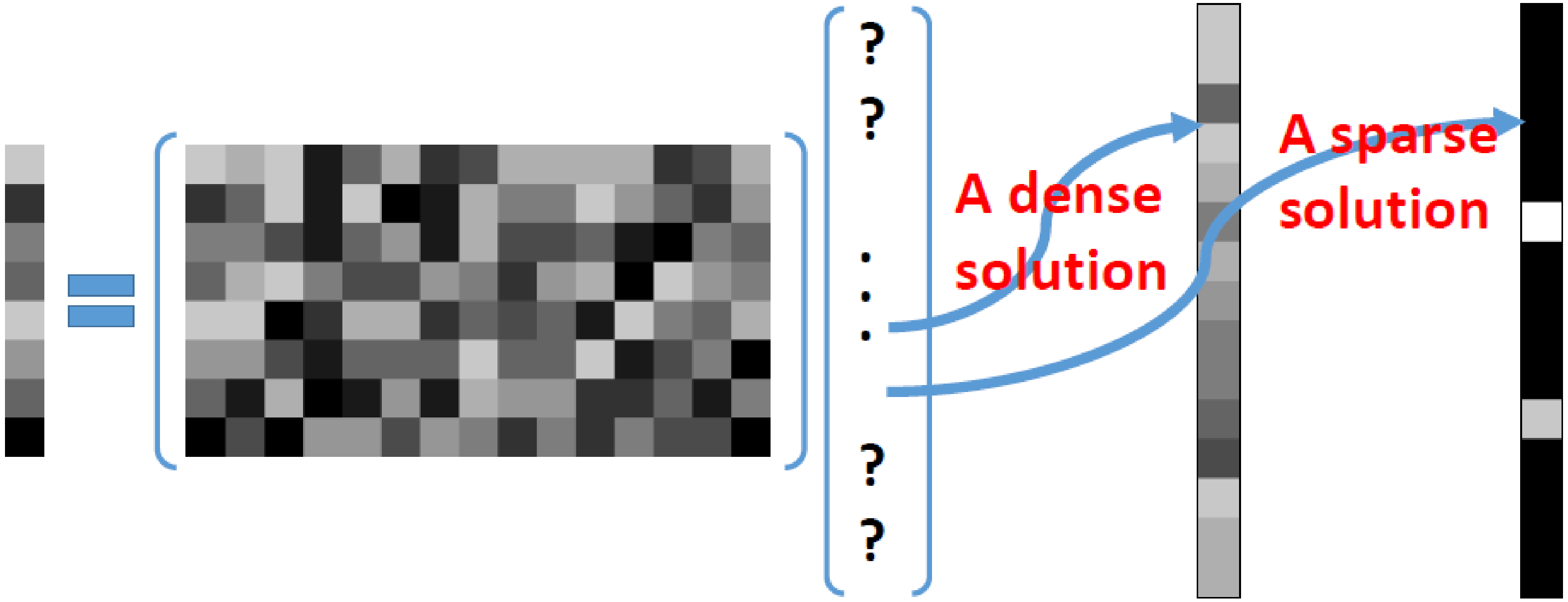
$$min_x \overset{\text{Fidelity}}{F(\boldsymbol{x},\boldsymbol{y})} + \lambda \cdot \overset{\text{Regularization (Prior)}}{R(\boldsymbol{x})}$$

Based on the image degradation process and the available image priors, build a model (objective function) and optimize it to estimate the latent image.

Sparse Representation for image restoration

A model for sparse solution

$$\min_{\alpha} \|\boldsymbol{\alpha}\|_0 \ s.t. \boldsymbol{A\alpha} = \boldsymbol{b}$$



- The matrix **A** is a fat matrix (underdetermined system)
- There is no solution in general
- The dense solution may not be useful or effective
- For more robust, we may need a sparse solution that has many zero entries

Image restoration: the problem and applications

A convex model

$$min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0 \; s.t. \boldsymbol{A\alpha} = \boldsymbol{b}$$

L_0-norm minimization is non-convex and NP-hard.

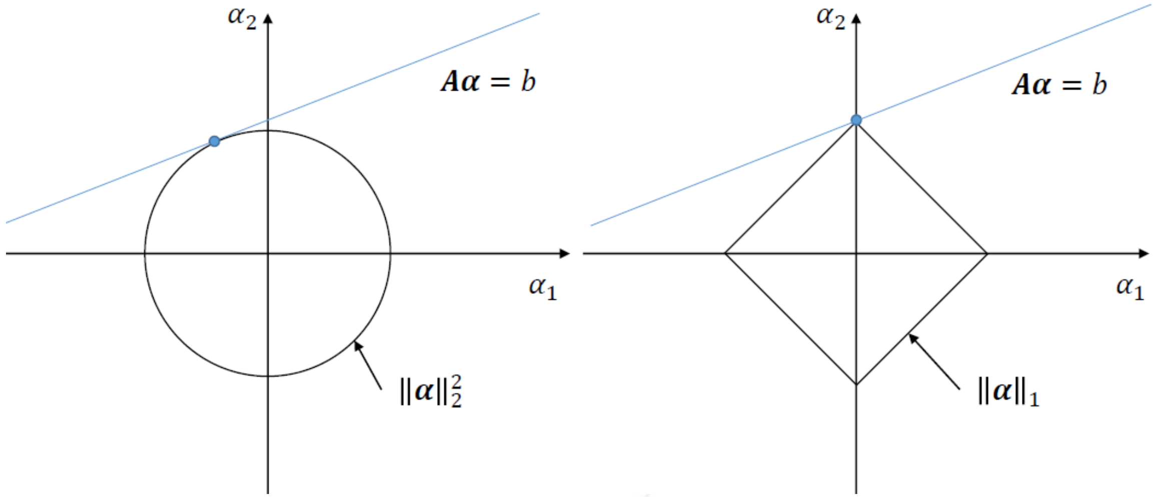
↓

$$min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1 \; s.t. \boldsymbol{A\alpha} = \boldsymbol{b}$$

L_1-norm minimization is tightest convex relaxation of L_0-norm minimization .

L_2-norm vs. L_1-norm

- Geometric illustration



A relaxed L_1 sparse coding model

$$min_{\alpha} ||\boldsymbol{A\alpha} - \boldsymbol{b}||_2^2 + \lambda||\boldsymbol{\alpha}||_1$$

- This is the most widely used sparse coding model, which is easy to solve and usually leads to a sparse solution.

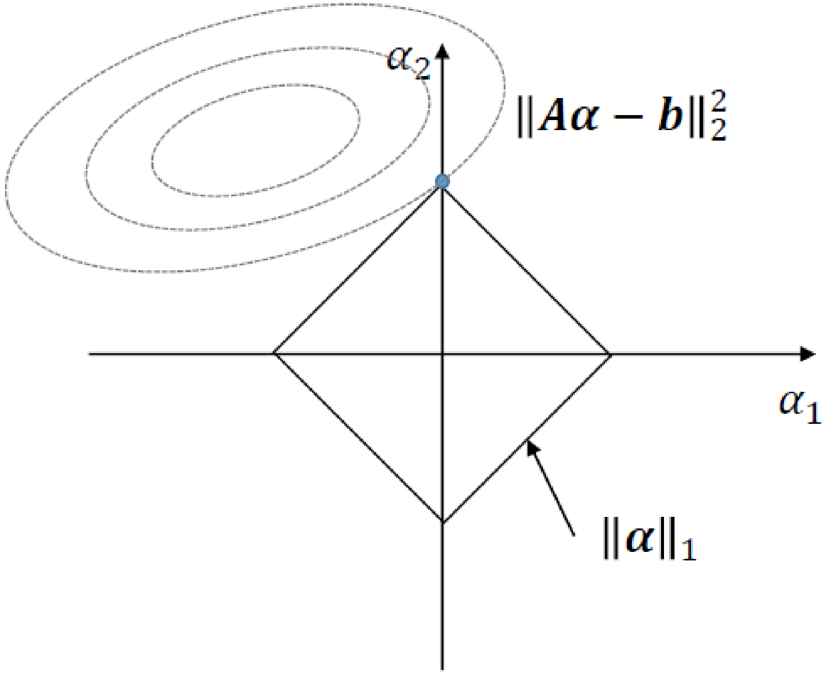

α_2
$||\boldsymbol{A\alpha} - \boldsymbol{b}||_2^2$
α_1
$||\boldsymbol{\alpha}||_1$

Image restoration: the problem and applications

How to adopt sparse coding for image restoration?

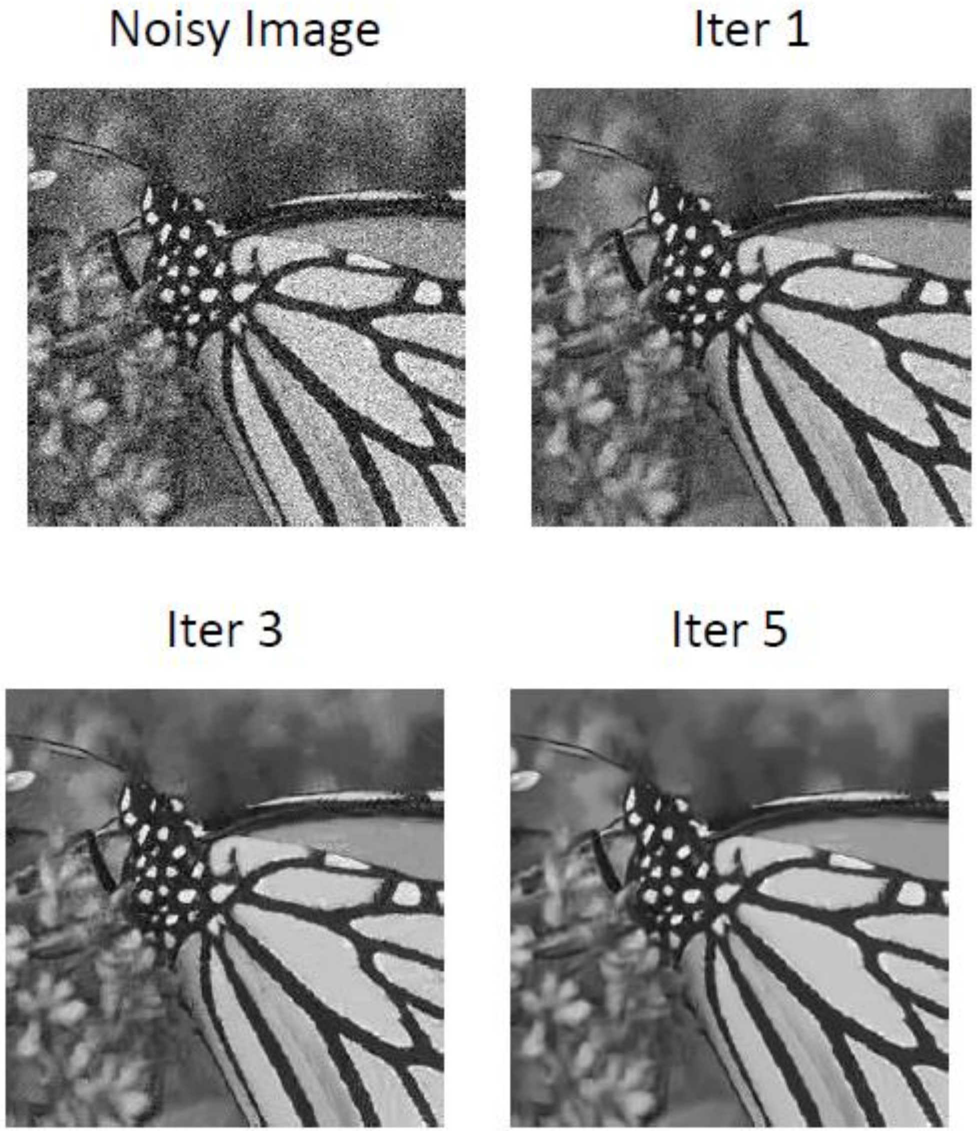
- Represent (encode) $\boldsymbol{x}$ over a dictionary $\boldsymbol{D}$, while enforcing the representation vector to be sparse:

$$min_{\alpha}\ \|\boldsymbol{\alpha}\|_1\ s.t.\ \boldsymbol{x}=\boldsymbol{D\alpha}$$

- Together with $min_{x}\ \|\boldsymbol{Hx}-\boldsymbol{y}\|_2^2+R(\boldsymbol{x})$, we have:

$$min_{\alpha}\ \|\boldsymbol{HD\alpha}-\boldsymbol{y}\|_2^2+\lambda\|\boldsymbol{\alpha}\|_1$$

- Solving $\boldsymbol{x}$ turns to solving $\boldsymbol{\alpha}$.



Why sparse: Bayesian perspective

- Signal recovery in a Bayesian viewpoint

$$\hat{\boldsymbol{x}} = argmax_{\boldsymbol{x}}\, P(\boldsymbol{x}|\boldsymbol{y}) \propto argmax_{\boldsymbol{x}}\, \overset{\text{Likelihood}}{P(\boldsymbol{y}|\boldsymbol{x})}\overset{\text{Prior}}{P(\boldsymbol{x})}$$

- Sparse Representation

$$\boldsymbol{x} = \boldsymbol{D\alpha}$$

- Prior: Assume that the representation coefficients follow some exponential distribution

$$\boldsymbol{\alpha}\sim exp\left(-\sum_i \|\alpha_i\|_p\right)$$

Why sparse: Bayesian perspective

$$\hat{\boldsymbol{x}} = argmax_{\boldsymbol{x}}\, P(\boldsymbol{x}|\boldsymbol{y}) \propto argmax_{\boldsymbol{x}}\, \overset{\text{Likelihood}}{P(\boldsymbol{y}|\boldsymbol{x})}\overset{\text{Prior}}{P(\boldsymbol{x})}$$

- The MAP solution:

$$\begin{aligned}\hat{\boldsymbol{\alpha}} &= argmax_{\boldsymbol{\alpha}}\, P(\boldsymbol{\alpha}|\boldsymbol{y})\\ &= argmax_{\boldsymbol{\alpha}}\, -log\, P(\boldsymbol{y}|\boldsymbol{\alpha}) - logP(\boldsymbol{\alpha})\\ &= argmin_{\boldsymbol{\alpha}}\, \|\boldsymbol{HD\alpha} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_p\end{aligned}$$

- If p=0， it is the L0 norm sparse coding problem
- If p=1， it becomes the convex L1 norm coding problem
- If p=2， it becomes the convex L2 norm problem， which generally has a close-form solution
- ……

Why sparsity helps signal recovery?

- A toy example:
  - ✓ If you are able to find your another half from all candidates all over the world (i.e., a large enough dictionary) , there is a very high probability (nearly 1) that you will find the one.

- An over-complete dictionary contains almost all possible under a specified scenario. A sparse solution with an over-complete dictionary often works!

- Sparsity (coefficients) and redundancy (dictionary) are the two sides of the same coin.

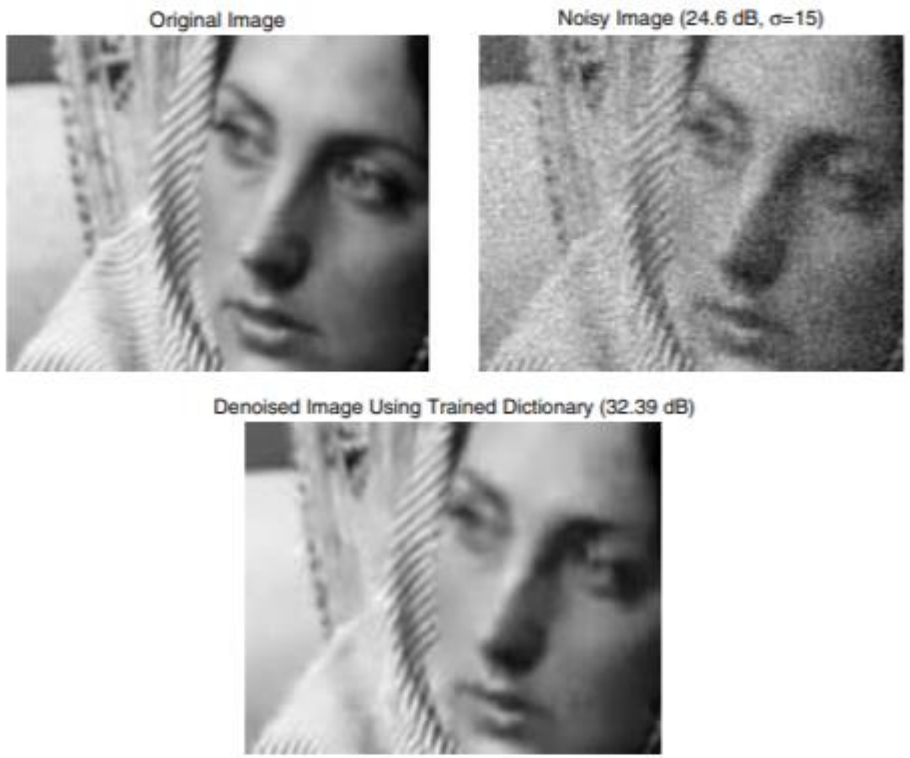
How to obtain a good dictionary? Learning!!!

- Sparse models with a learned over-complete dictionary often work better than analytically designed dictionaries such as DCT dictionary and wavelet dictionary.

Why learning

- More adaptive to specific task/data.
- Less strict constraints on the mathematical properties of basis (dictionary atom).
- More flexible to model data.
- Tend to produce sparser solutions to many problems.

Applications of IR with Sparse Representation: Denoising



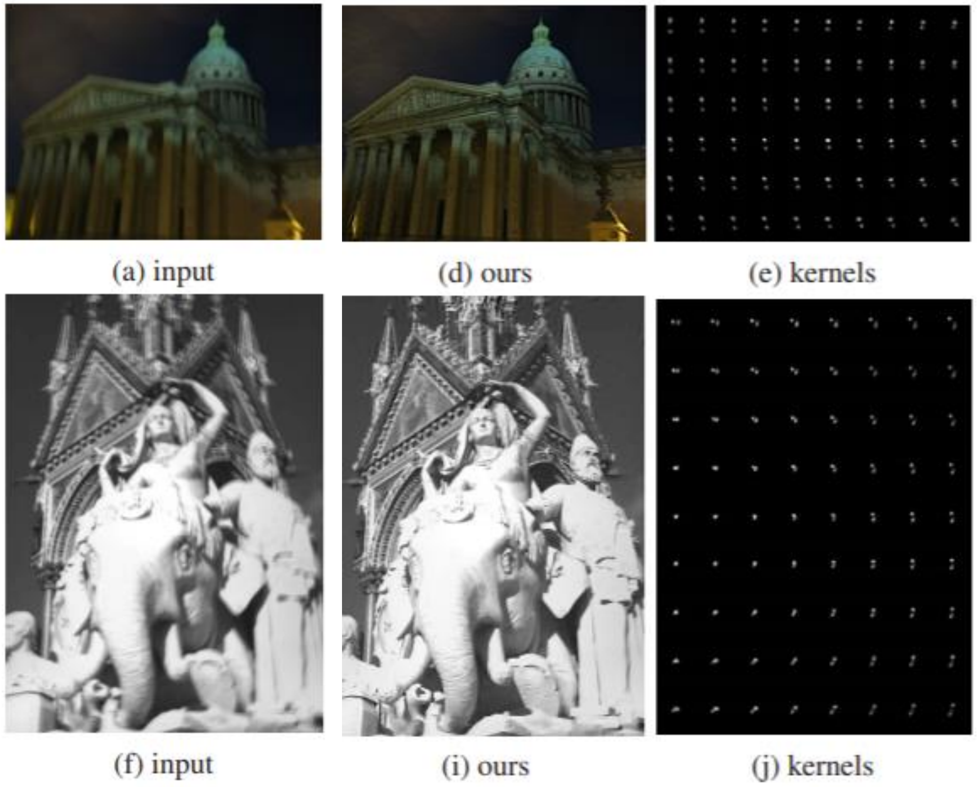
Original Image

Noisy Image (24.6 dB, σ=15)

Denoised Image Using Trained Dictionary (32.39 dB)

- **KSVD**
  Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries (TIP 2006), Elad et al.
- **LSSC**
  Non-local Sparse Models for Image Restoration (ICCV 2009), Mairal et al.
- **NCSR**
  Nonlocally Centralized Sparse Representation for Image Restoration (TIP 2012), Dong et al.
- **OCTOBOS**
  Structured Overcomplete Sparsifying Transform Learning with Convergence Guarantees and Applications (IJCV 2015), Wen et al.
- **GSR**
  Group-based Sparse Representation for Image Restoration (TIP 2014), Zhang et al.
- **TWSCA**
  Trilateral Weighted Sparse Coding Scheme for Real-World Image Denoising (ECCV 2018), Xu et al.

Applications of IR with Sparse Representation: Deblurring

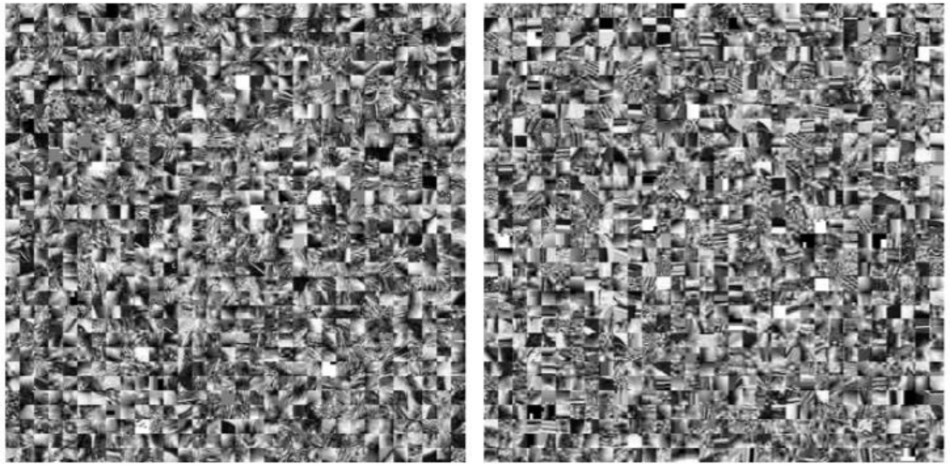


(a) input (d) ours (e) kernels

(f) input (i) ours (j) kernels

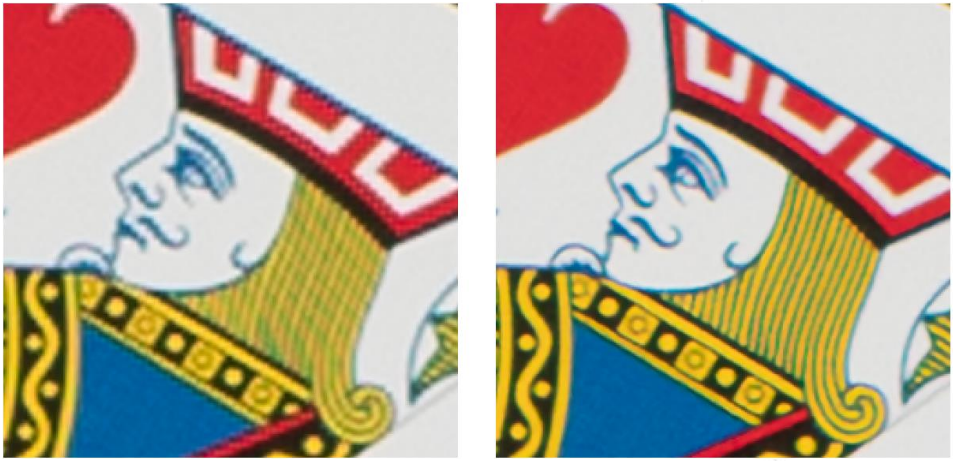
- Spatially-varying out-of-focus image deblurring with L1-2 optimization and a guided blur map
- Unnatural L0 sparse representation for natural image deblurring
- Deblurring Text Images via L0 -Regularized Intensity and Gradient Prior
- Non-Uniform Camera Shake Removal Using a Spatially-Adaptive Sparse Penalty
- Fast Non-Blind Image De-blurring With Sparse Priors
- Multi-image Blind Deblurring Using a Coupled Adaptive Sparse Prior

Applications of IR with Sparse Representation: Super-resolution



(a) Foreground Dict

(b) Background Dict



•ScSR [Web]

•Image super-resolution as sparse representation of raw image patches (CVPR2008), Jianchao Yang et al.

•Image super-resolution via sparse representation (TIP2010), Jianchao Yang et al.

•Coupled dictionary training for image super-resolution (TIP2011), Jianchao Yang et al.

•Deep Networks for Image Super-Resolution with Sparse Prior (ICCV2015), Zhaowen Wang et al.

•Robust Single Image Super-Resolution via Deep Networks with Sparse Prior (TIP2016), Ding Liu et al.

•VDSR [Web] [Unofficial Implementation in Caffe]

Questions on Sparse Representation

- What is a good dictionary?
- How to learn a good dictionary?
- ……

Low-rank minimization for image restoration

Let’s start from PCA

- Principal components analysis (PCA) is one of a family of techniques for taking high-dimensional data, and using the dependencies between the variables to represent it in a more tractable, lower-dimensional form, without losing too much information.

- It transforms the data to a new coordinate system such that the greatest variance by some scalar projection of the data comes to lie on the first coordinate (called the first principal component), the second greatest variance on the second coordinate, and so on.

- In order to maximize variance, the first weight vector w(1) thus has to satisfy

$$\mathbf{w}_{(1)} = \underset{\|\mathbf{w}\|=1}{\arg\max}\left\{\sum_i (t_1)^2_{(i)}\right\} = \underset{\|\mathbf{w}\|=1}{\arg\max}\left\{\sum_i \left(\mathbf{x}_{(i)}\cdot\mathbf{w}\right)^2\right\}$$

Image restoration: the problem and applications

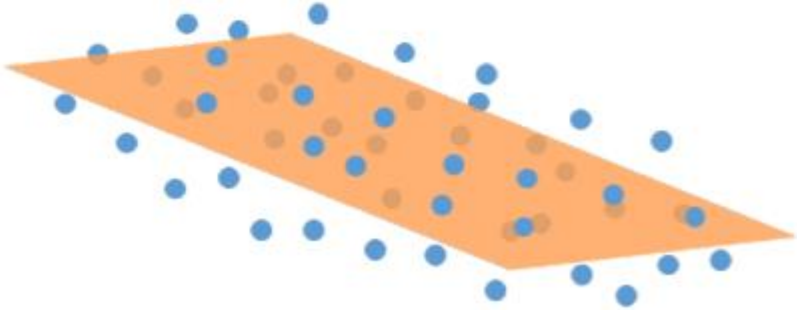
From the perspective of low-rank

- N samples $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{n \times N}$ that are centered
- PCA: seeks r directions that explain most variance of data

$$\text{minimize}_{\boldsymbol{L}:\text{rank}(\boldsymbol{L})=r} \quad \|\boldsymbol{X} - \boldsymbol{L}\|_{\text{F}}$$

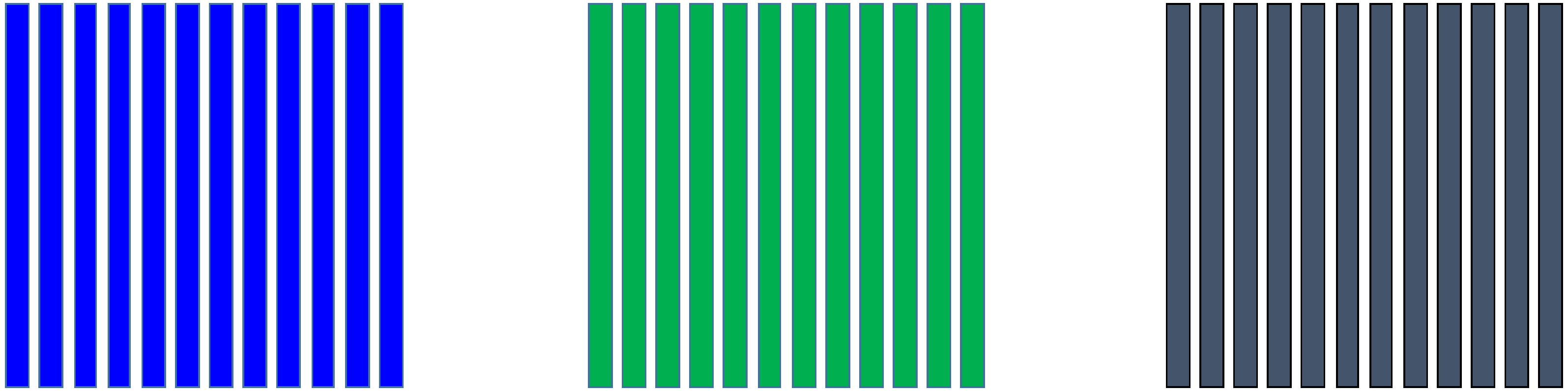
$$\begin{array}{ll} \text{minimize} & \|M - L\| \\ \text{subject to} & \text{rank}(L) \le k. \end{array}$$

- ○ best rank-r approximation of $\boldsymbol{X}$



Data representation

$$\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{E}$$



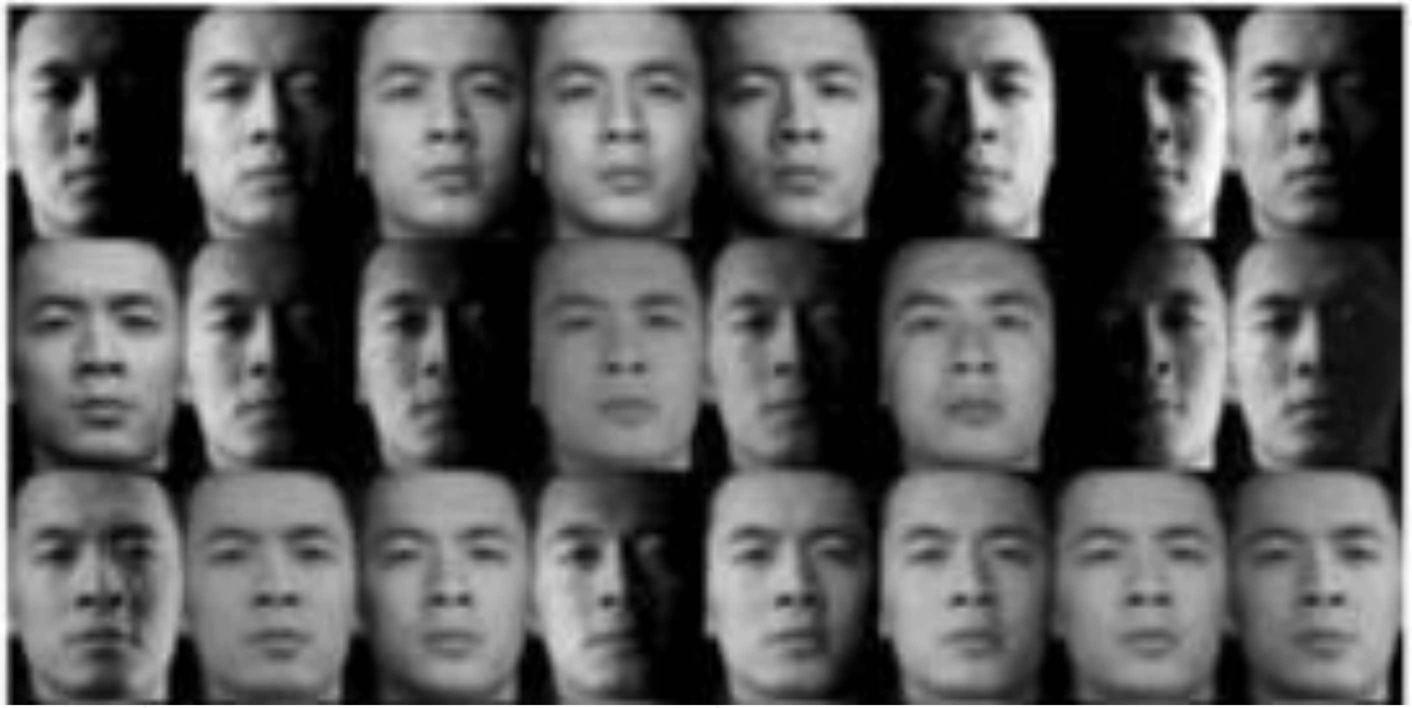
Each column corresponds to a sample

The desired latent low-rank matrix

The residual matrix

Image restoration: the problem and applications

Visual data often has an intrinsic low rank structure



Nuclear norm minimization

Considering the fact (i.e., prior) that the input vectors are highly correlated, we can take them as a 2D low rank matrix and minimize its rank:

$$Rank(\boldsymbol{X}) = \sum \|\sigma_i(\boldsymbol{X})\|_0$$

The above rank function is non-convex. A convex relaxation of it is the so-called nuclear norm:

$$\|\boldsymbol{X}\|_* = \sum \|\sigma_i(\boldsymbol{X})\|_1$$

Nuclear norm minimization

Nuclear norm minimization (NNM) can be used to estimate the latent low rank matrix $\boldsymbol{X}$ form $\boldsymbol{Y}$ via the following unconstrained minimization problem:

$$\hat{\boldsymbol{X}} = argmin_{\boldsymbol{X}} \|\boldsymbol{Y} - \boldsymbol{X}\|_F^2 + \lambda \|\boldsymbol{X}\|_*$$

Closed form solution (Cai, et al., SIAM10)

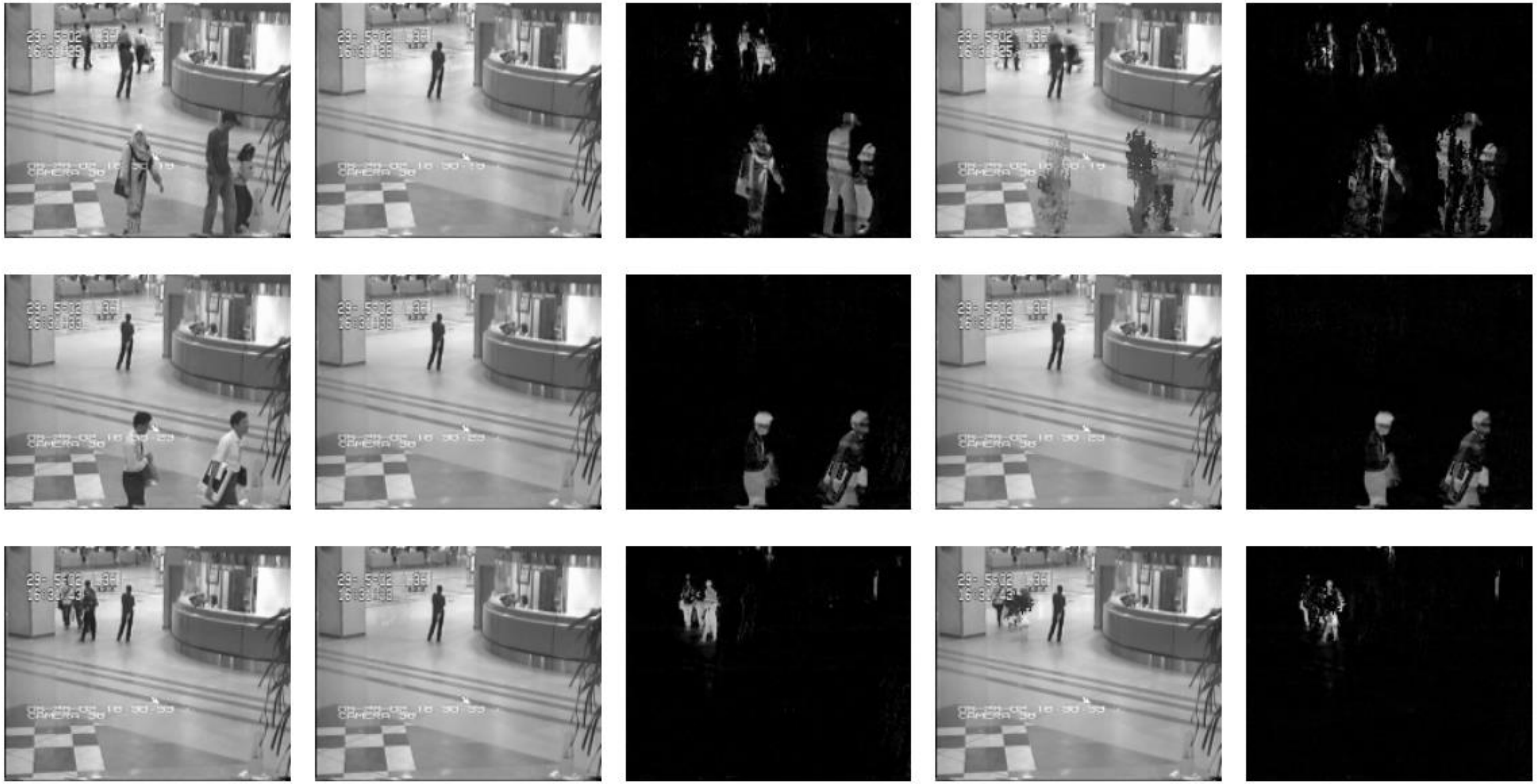
$$\hat{\boldsymbol{X}} = \boldsymbol{U} S_\lambda(\boldsymbol{\Sigma}) \boldsymbol{V}^T$$

where $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ is the SVD of $\boldsymbol{Y}$, and

$$S_\lambda(\boldsymbol{\Sigma})_{ii} = \max\left(\boldsymbol{\Sigma}_{ii} - \frac{\lambda}{2}, 0\right)$$

Image restoration: the problem and applications

Background modeling

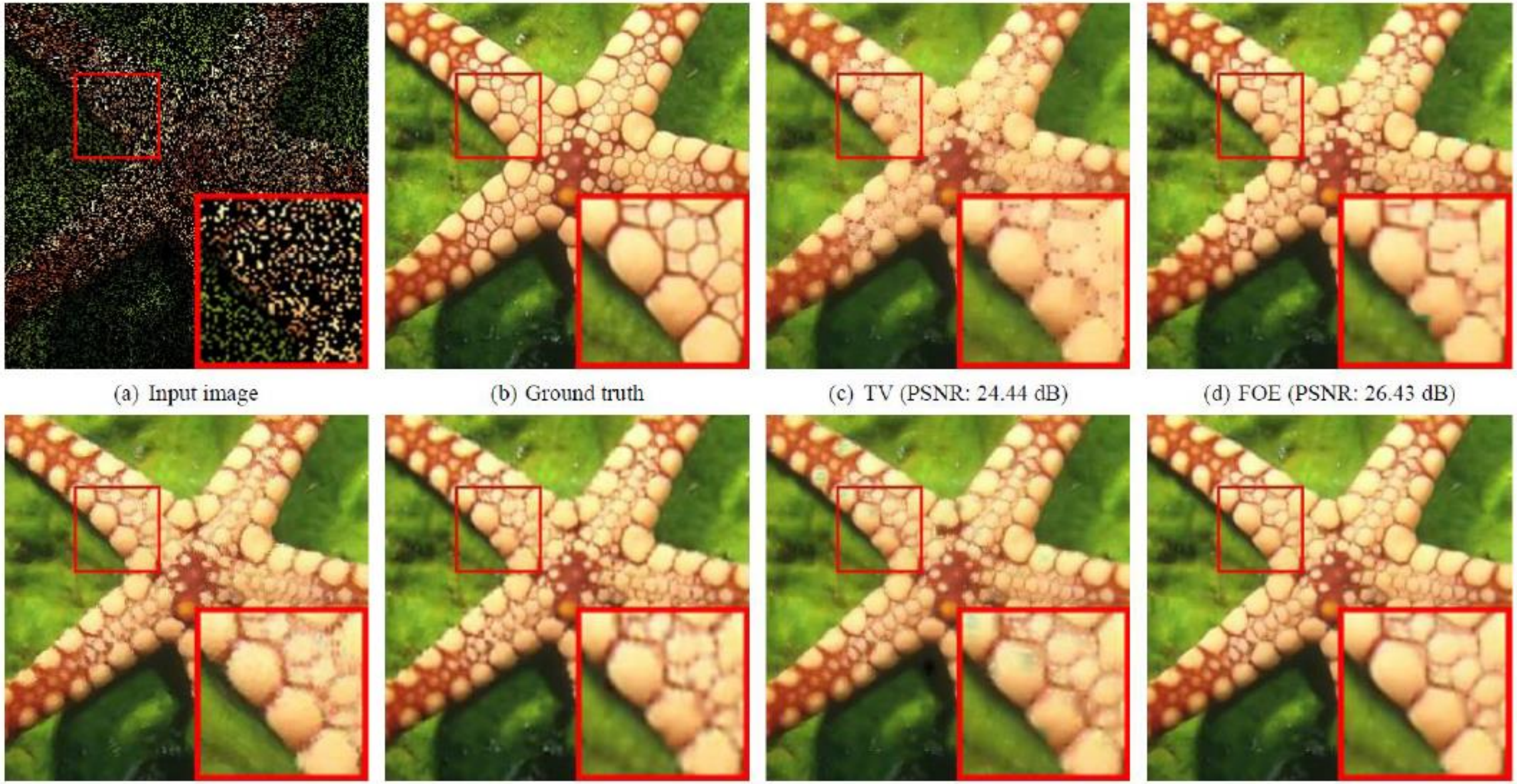


(a) Original frames (b) Low-rank $\hat{L}$ (c) Sparse $\hat{S}$ (d) Low-rank $\hat{L}$ (e) Sparse $\hat{S}$

Convex optimization (this work) Alternating minimization [47]

Image restoration: the problem and applications

Image Inpainting



(a) Input image

(b) Ground truth

(c) TV (PSNR: 24.44 dB)

(d) FOE (PSNR: 26.43 dB)

(e) VNL (PSNR: 24.36 dB)

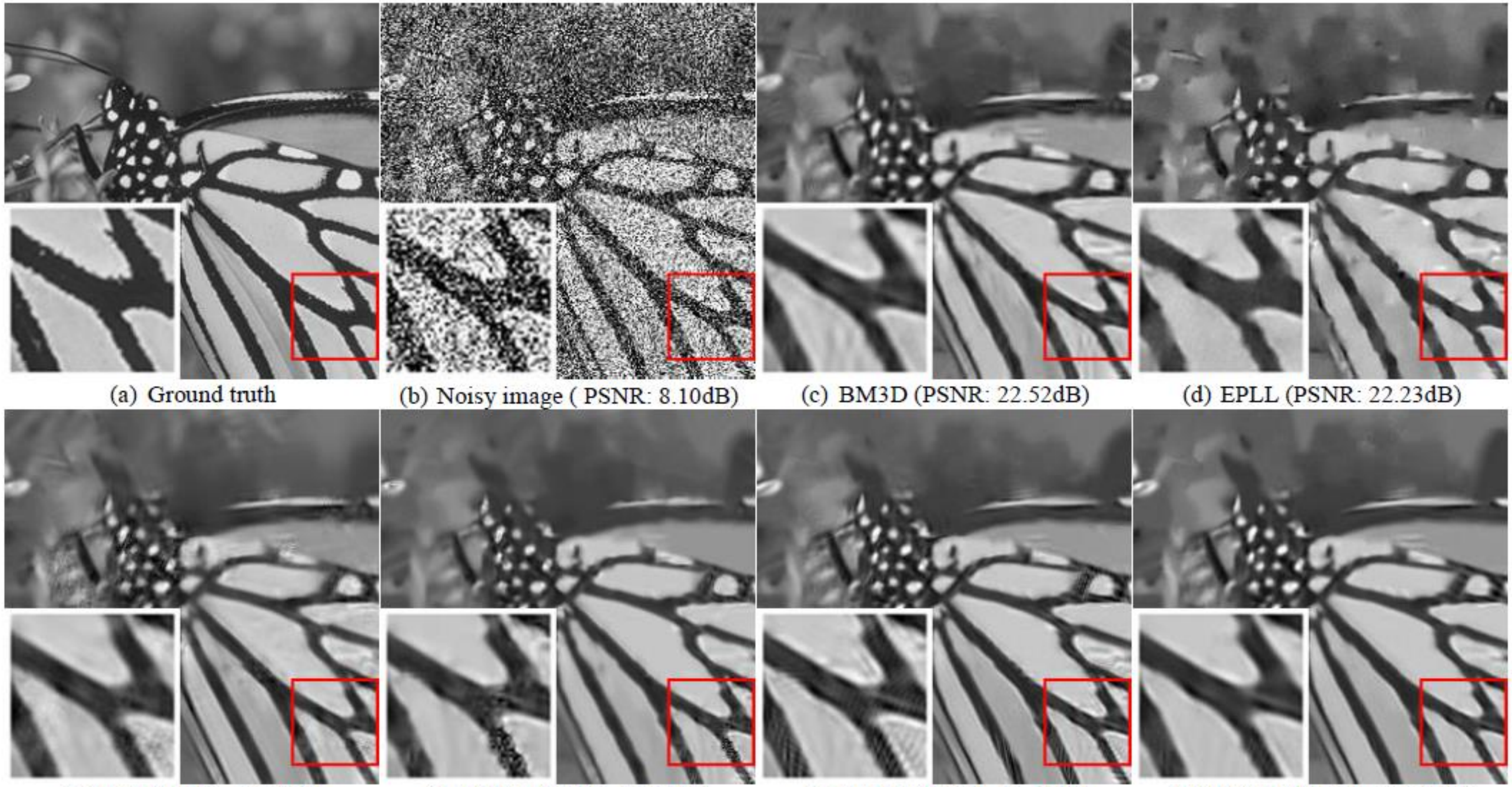
(f) BPDL (PSNR: 26.57 dB)

(g) NNM (PSNR: 25.45 dB)

(h) WNNM (PSNR: 27.11 dB)

Image restoration: the problem and applications

Image Denoising



(a) Ground truth

(b) Noisy image (PSNR: 8.10dB)

(c) BM3D (PSNR: 22.52dB)

(d) EPLL (PSNR: 22.23dB)

(e) SSC (PSNR: 22.24dB)

(f) NCSR (PSNR: 22.11dB)

(g) SAIST (PSNR: 22.61dB)

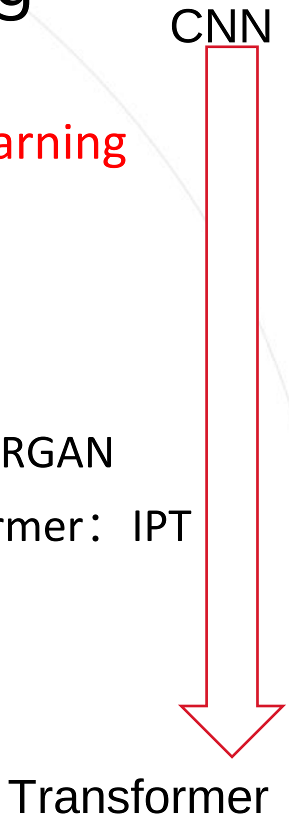
(h) WNNM (PSNR: 22.91dB)

Deep learning for image restoration

Image restoration: the problem and applications

- Methodology overview: from classic computer vision to deep learning

- Filtering based methods
  - Gaussian filtering
  - PDE-based anisotropic diffusion
  - Bilateral filtering
  - Nonlocal means filtering
  - Domain transfer based filtering
  - ……
- Transform based methods
  - Fourier transform
  - Wavelet transform
  - Curvelet transform
  - Ridgelet transform
  - Bandlet transform
  - ……
- Model based optimization
  - Regularization based
  - Sparse representation
  - Low-rank minimization
  - ……
- Deep learning
  - SRCNN
  - VDSR
  - ESPCNN
  - GAN：SRGAN
  - Transformer：IPT
  - ……



CNN

Transformer

Image restoration: the problem and applications

- Learn a compact inference or a mapping function from a training set of degraded-latent image pairs.
- General formulation:

$$min_{\Theta}\overset{\text{Loss function}}{\overset{\uparrow}{loss}}(\hat{\boldsymbol{x}},\boldsymbol{x})\quad s.t.\hat{\boldsymbol{x}}=F(\boldsymbol{y},\boldsymbol{H};\overset{\text{Set of parameters to be learned}}{\overset{\uparrow}{\Theta}})$$

- Key issues
  - The availability of paired training data
  - The design of learning architecture
  - The definition of loss function

Why deep learning?

- Strong learning capacity

  End-to-end learning for the inference/mapping function

  Deeper architecture for strong and distinct image priors

- Architecture design

  Residual learning or other structures

  Batch normalization and other network regularizations

  Various blocks, e.g., Conv, Deconv, Pooling, …

- Optimization algorithms

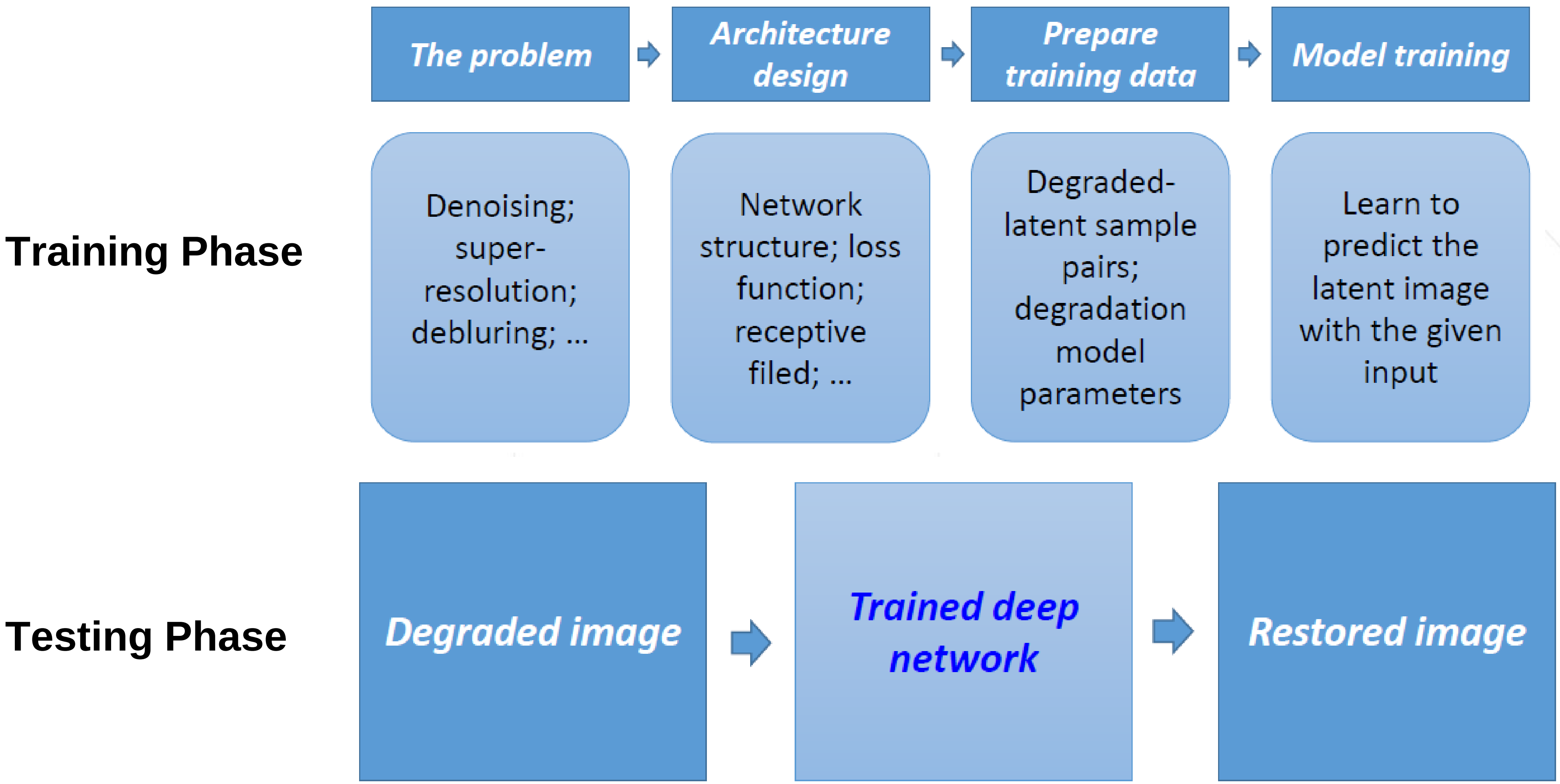
  SGD, momentum SGD, Adam,……

- Speed

  GPU/NPU/DSP

Image restoration: the problem and applications

General pipeline



Training Phase

The problem → *Architecture design* → *Prepare training data* → *Model training*

Denoising; super-resolution; debluring; …

Network structure; loss function; receptive filed; …

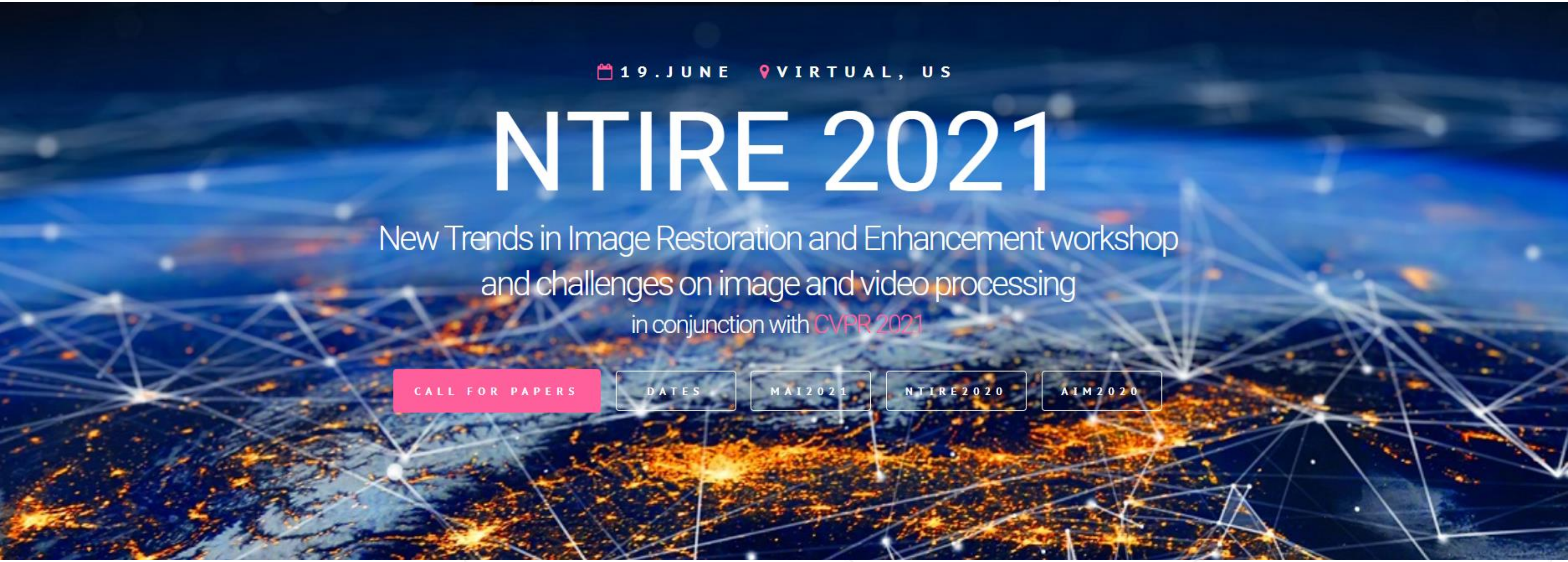
Degraded-latent sample pairs; degradation model parameters

Learn to predict the latent image with the given input

Testing Phase

Degraded image → *Trained deep network* → *Restored image*

Challenge：New Trends in Image Restoration and Enhancement



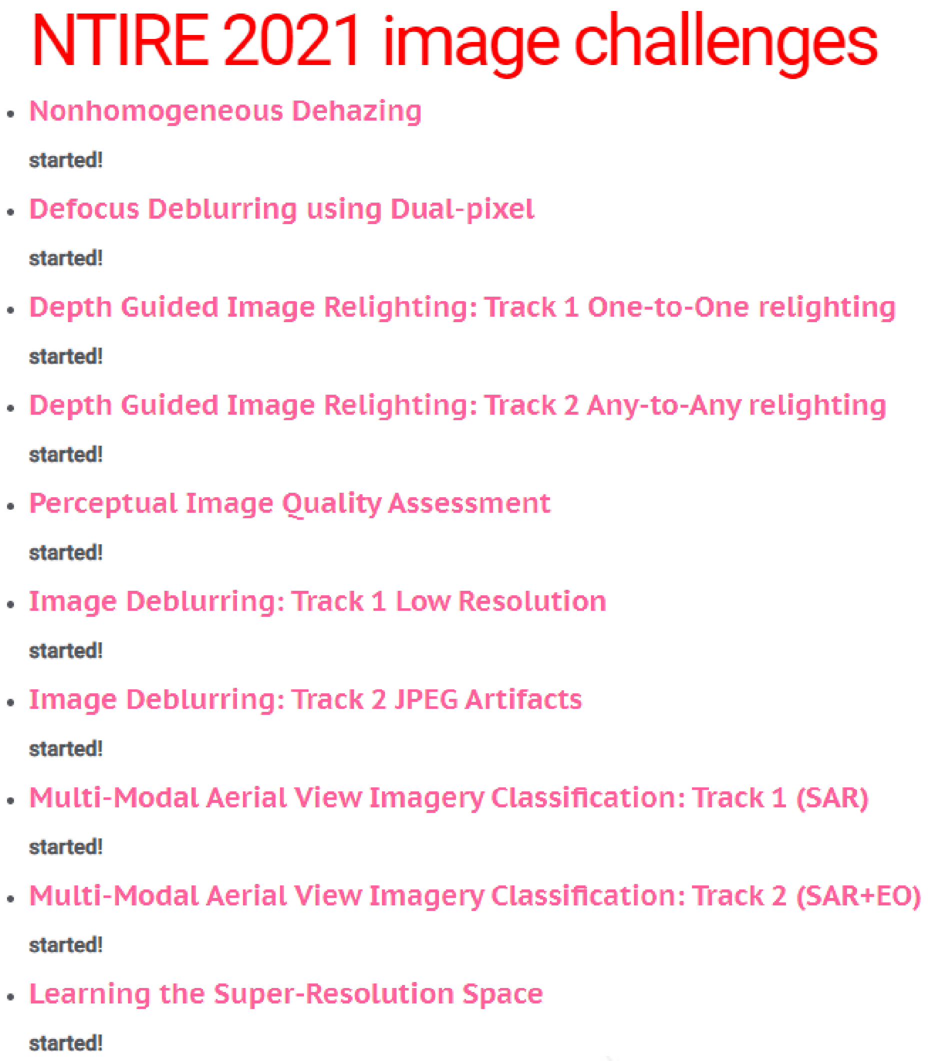
19.JUNE VIRTUAL, US

NTIRE 2021

New Trends in Image Restoration and Enhancement workshop and challenges on image and video processing in conjunction with CVPR 2021

Image restoration: the problem and applications

Challenge： New Trends in Image Restoration and Enhancement



NTIRE 2021 image challenges

- Nonhomogeneous Dehazing

  started!
- Defocus Deblurring using Dual-pixel

  started!
- Depth Guided Image Relighting: Track 1 One-to-One relighting

  started!
- Depth Guided Image Relighting: Track 2 Any-to-Any relighting

  started!
- Perceptual Image Quality Assessment

  started!
- Image Deblurring: Track 1 Low Resolution

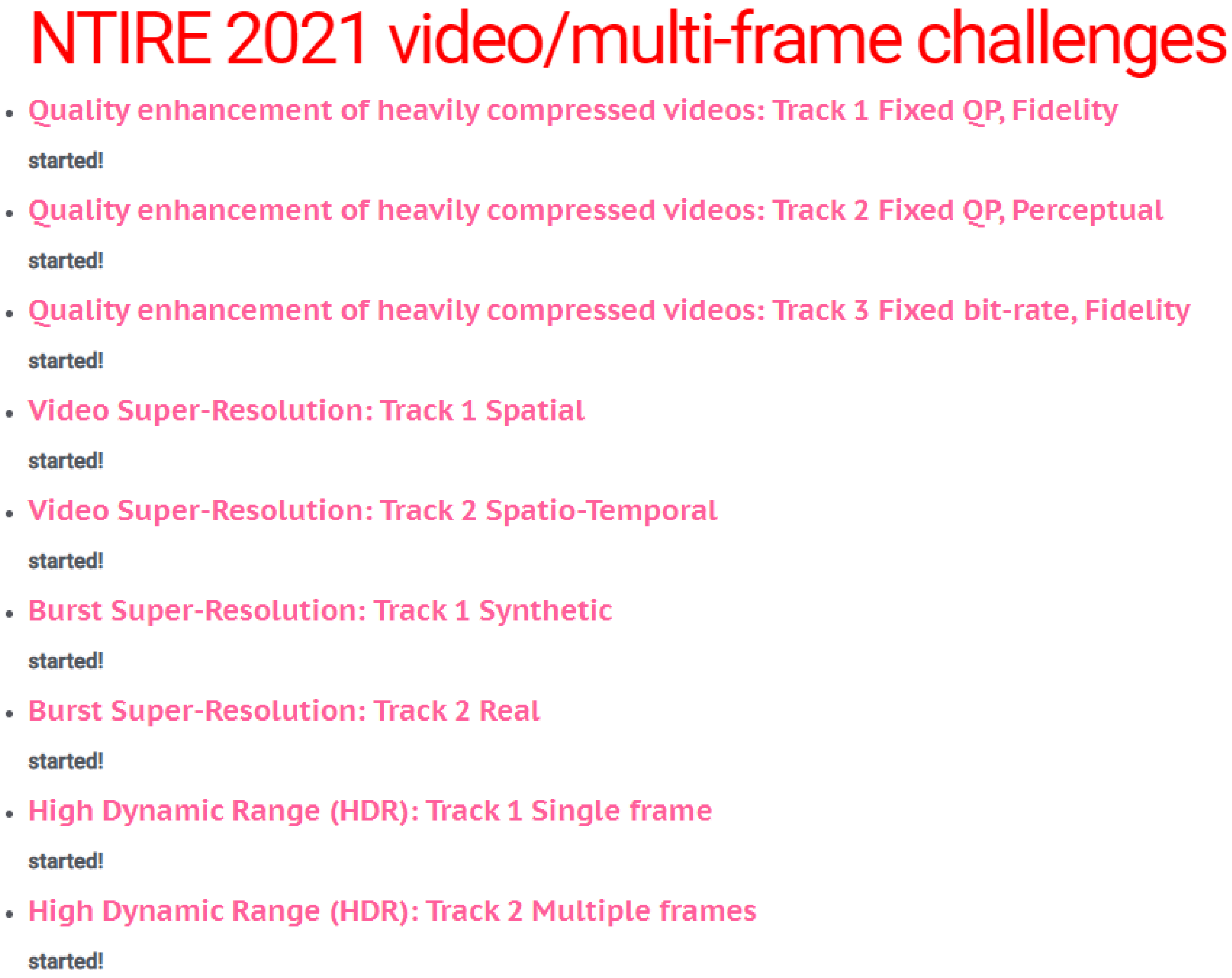
  started!
- Image Deblurring: Track 2 JPEG Artifacts

  started!
- Multi-Modal Aerial View Imagery Classification: Track 1 (SAR)

  started!
- Multi-Modal Aerial View Imagery Classification: Track 2 (SAR+EO)

  started!
- Learning the Super-Resolution Space

  started!



NTIRE 2021 video/multi-frame challenges

- Quality enhancement of heavily compressed videos: Track 1 Fixed QP, Fidelity

  started!
- Quality enhancement of heavily compressed videos: Track 2 Fixed QP, Perceptual

  started!
- Quality enhancement of heavily compressed videos: Track 3 Fixed bit-rate, Fidelity

  started!
- Video Super-Resolution: Track 1 Spatial

  started!
- Video Super-Resolution: Track 2 Spatio-Temporal

  started!
- Burst Super-Resolution: Track 1 Synthetic

  started!
- Burst Super-Resolution: Track 2 Real

  started!
- High Dynamic Range (HDR): Track 1 Single frame

  started!
- High Dynamic Range (HDR): Track 2 Multiple frames

  started!

Super-resolution via CNN (SRCNN)

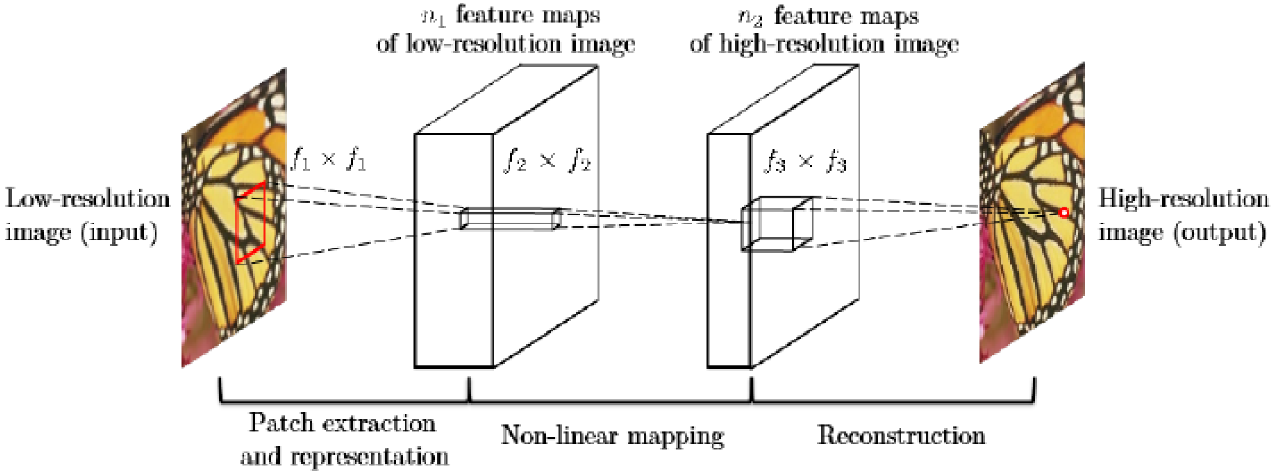


Fig. 2. Given a low-resolution image $\mathbf{Y}$, the first convolutional layer of the SRCNN extracts a set of feature maps. The second layer maps these feature maps nonlinearly to high-resolution patch representations. The last layer combines the predictions within a spatial neighbourhood to produce the final high-resolution image $F(\mathbf{Y})$.

256×256 (input, bicubic interpolation) → 256 × 256 × 64 (feature map of Conv1) → 256 × 256 × 32 (feature map of Conv2) → 256 × 256 (output)

Very deep CNN for SR (VDSR)

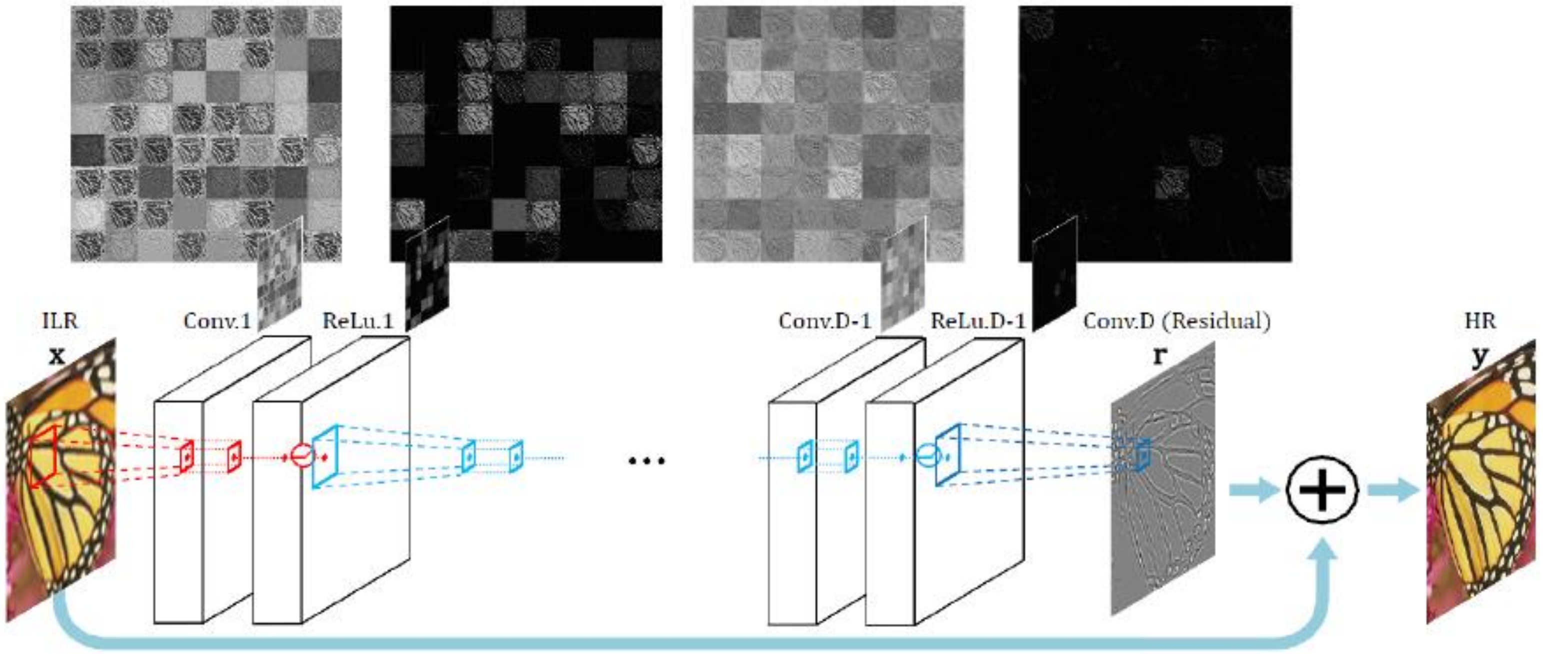
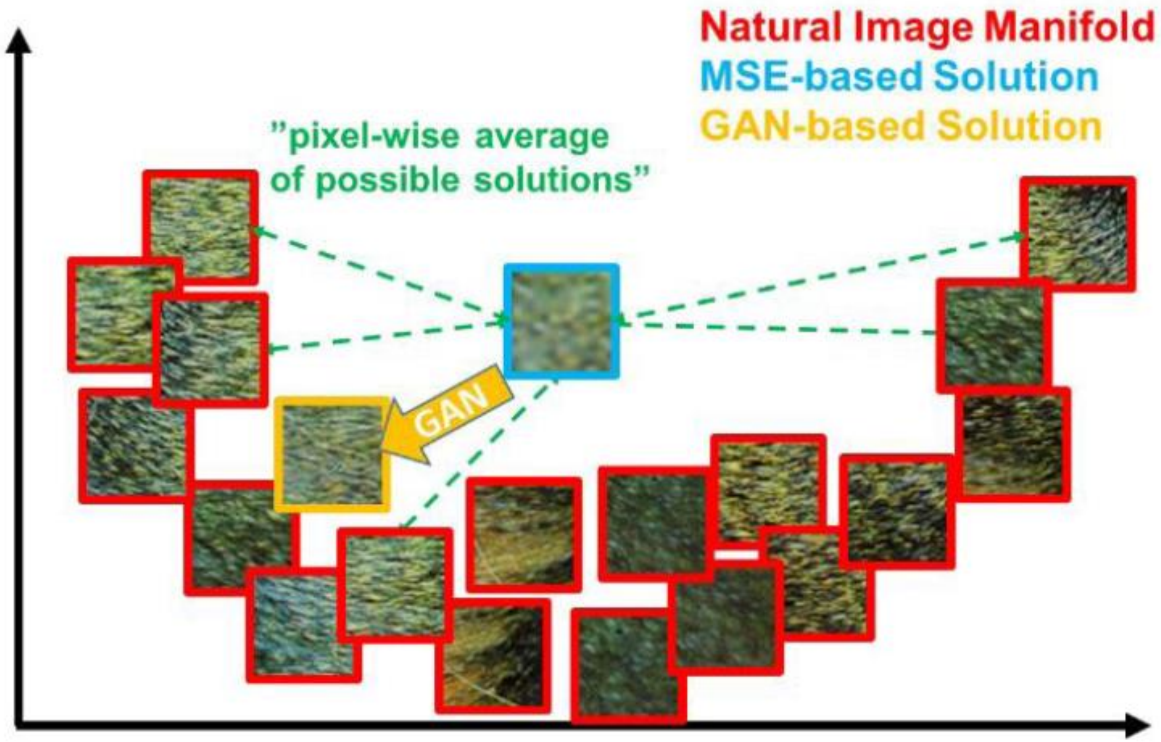


Figure 2: Our Network Structure. We cascade a pair of layers (convolutional and nonlinear) repeatedly. An interpolated low-resolution (ILR) image goes through layers and transforms into a high-resolution (HR) image. The network predicts a residual image and the addition of ILR and the residual gives the desired output. We use 64 filters for each convolutional layer and some sample feature maps are drawn for visualization. Most features after applying rectified linear units (ReLu) are zero.

SR by GAN (SRGAN): motivation



- *MSE-based solution appears overly smooth due to the pixel-wise average of possible solutions in the pixel space.*
- *Using GAN (Generative Adversarial Network) to drive the reconstruction towards the natural image manifold producing perceptually more convincing solutions.*

Image processing Transformer

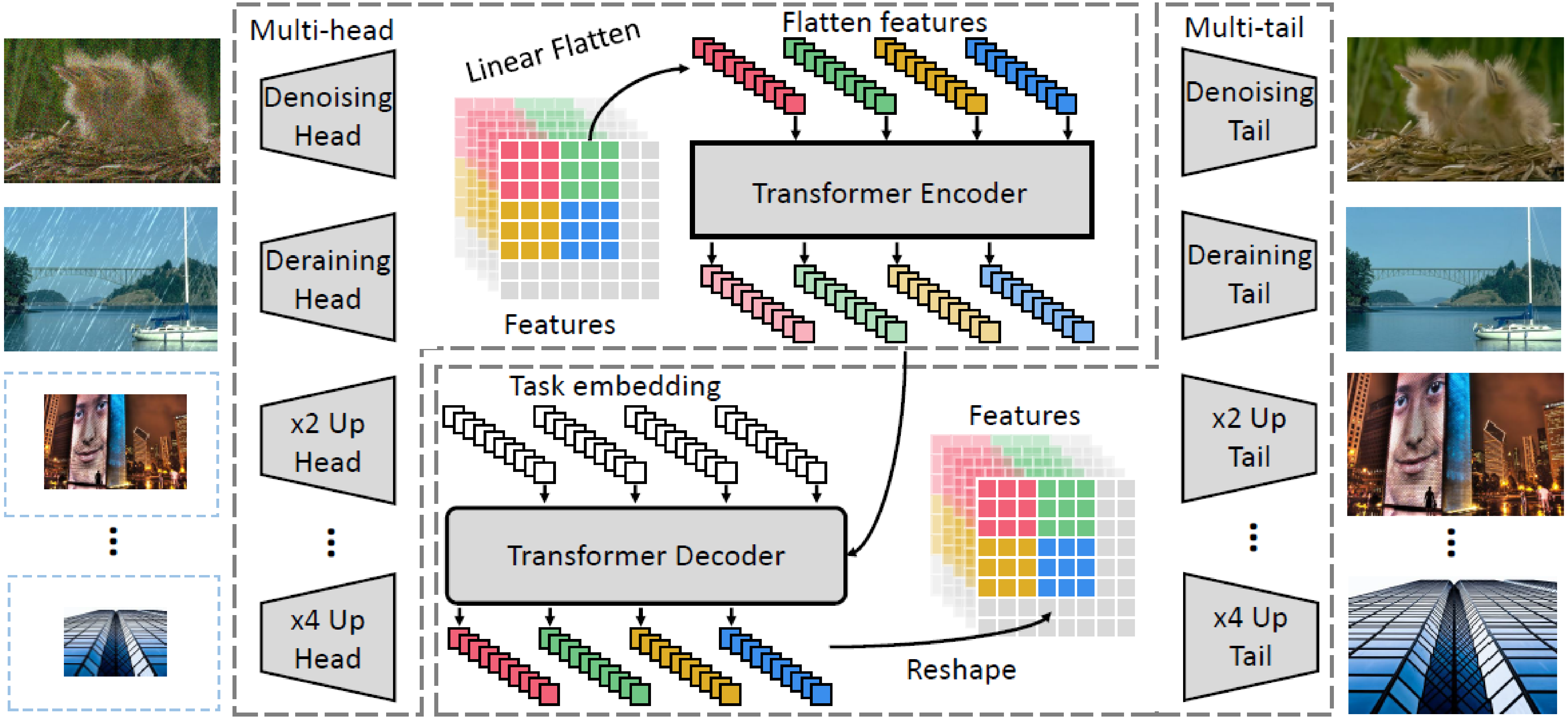


Figure 2. The diagram of the proposed image processing transformer (IPT). The IPT model consists of multi-head and multi-tail for different tasks and a shared transformer body including encoder and decoder. The input images are first converted to visual features and then divided into patches as visual words for subsequent processing. The resulting images with high visual quality are reconstructed by ensembling output patches.

商汤 sensetime

清華大学 Tsinghua University

Outline

Part 1	Image Restoration: Problems and Methods
Part 2	Generative Image Models for Image Restoration

商汤
sensetime

清華大學
Tsinghua University

Generative Image Models

Part 1	Why Generative Image Models
Part 2	Explicit Density Image Models
Part 3	Generative Adversarial Networks
Part 4	Generative Image Models Beyond GANs

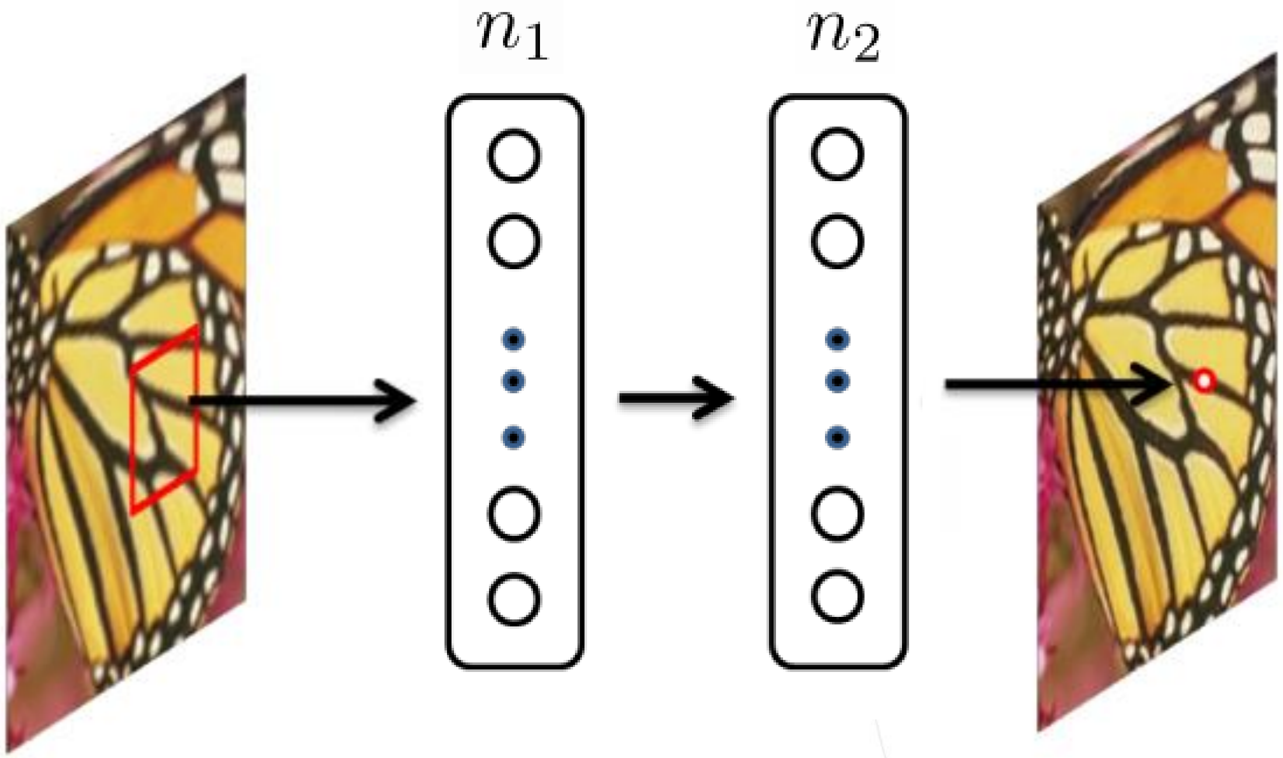
Introduction to Generative Image Models

Generative Image Models

- Machine learning models are either discriminative or generative
- Discriminative image models observe images, and map them to the data of another space
- Generative models observe the images, and learn how they are generated

Introduction to Generative Image Models

- Learning to Regress Target Images: Best Image Restoration Pipeline?



Loss $=\left\|I_{pred} - I_{gt}\right\|^2$

Image source: Learning a Deep Convolutional Network for Image Super-Resolution , ECCV 2014

Regressing Target Images DO NOT Learn Good Image Restoration

Assume we have a infinitely large dataset, such that every corrupted image has many and many possible groundtruth versions.

$$\begin{tikzcd}
 & & I_{gt1} \\
I_{input} \arrow[r] & I_{pred} \arrow[ur] \arrow[r] \arrow[dr] & I_{gt2} \\
 & & I_{gt3}
\end{tikzcd}$$

Introduction to Generative Image Models

- Regressing Target Images DO NOT Learn Good Image Restoration

Assume we have a infinitely large dataset, such that every corrupted image has many and many possible groundtruth versions.

What we minimize:

$$\int \left\| I_{pred} - I_{gt} \right\|^2 P\left(I_{gt} \middle| I_{input}\right)$$

$$\begin{tikzcd}
 & & I_{gt1} \\
I_{input} \arrow[r] & I_{pred} \arrow[ur] \arrow[r] \arrow[dr] & I_{gt2} \\
 & & I_{gt3}
\end{tikzcd}$$

Introduction to Generative Image Models

■ Regressing Target Images DO NOT Learn Good Image Restoration

Assume we have a infinitely large dataset, such that every corrupted image has many and many possible groundtruth versions.

What we minimize:

$$\int \left\| I_{pred} - I_{gt} \right\|^2 P\left(I_{gt} | I_{input}\right)$$

$$\begin{tikzcd}
 & & I_{gt1} \\
I_{input} \arrow[r] & I_{pred} \arrow[ur] \arrow[r] \arrow[dr] & I_{gt2} \\
 & & I_{gt3}
\end{tikzcd}$$

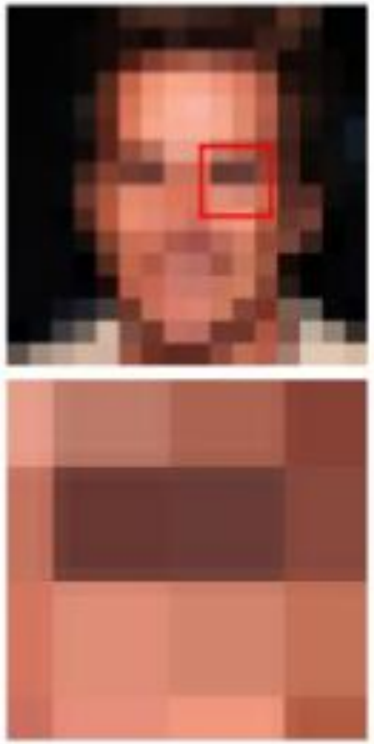
This is minimized by the weighted average of all groundtruths of I_{input}.

$$I_{pred} = \int I_{gt} P\left(I_{gt} | I_{input}\right)$$

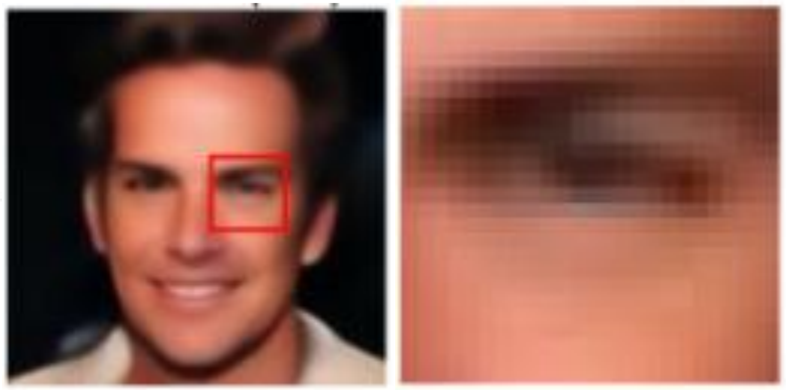
Photometric losses, such as L2 distance, tend to smooth out high-frequency image details.

Introduction to Generative Image Models

Learning to Sample, Instead of Learning to Regress



A deep CNN that learns statistical mean of input

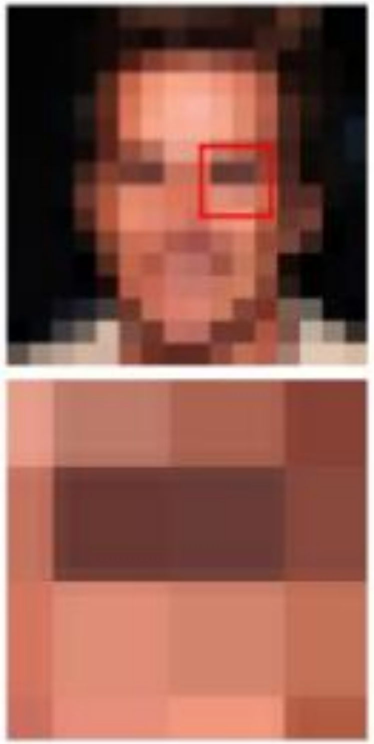


No high frequency details

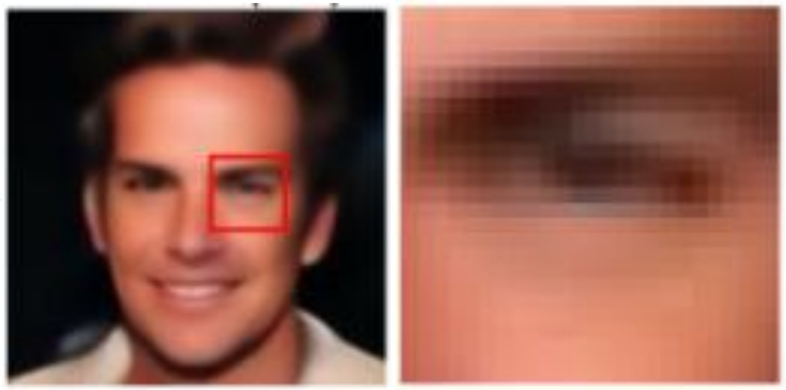
Image source: http://pulse.cs.duke.edu/, https://laptrinhx.com/how-to-train-stylegan-to-generate-realistic-faces-1888828401/

Introduction to Generative Image Models

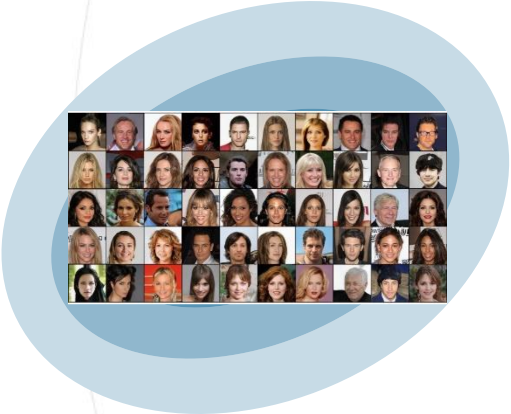
- Learning to Sample, Instead of Learning to Regress



A deep CNN that learns statistical mean of input



No high frequency details

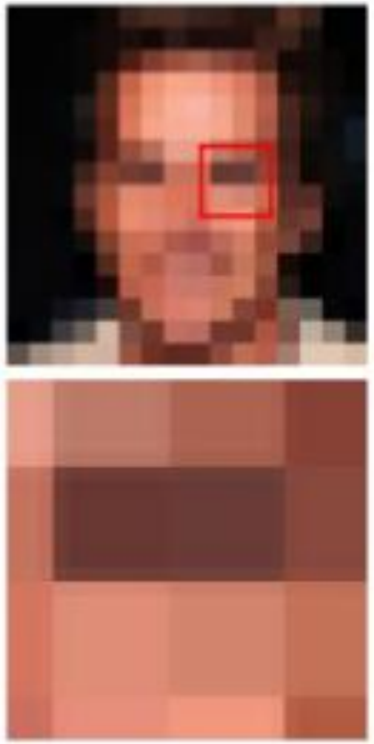


What is better: learn a distribution of real human faces

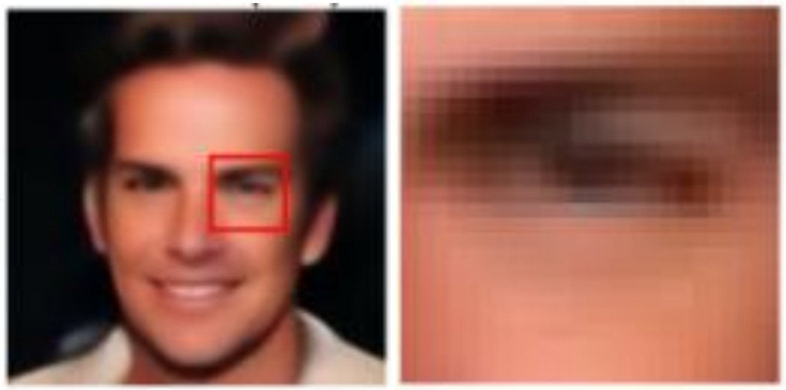
Image source: http://pulse.cs.duke.edu/, https://laptrinhx.com/how-to-train-stylegan-to-generate-realistic-faces-1888828401/

Introduction to Generative Image Models

- Learning to Sample, Instead of Learning to Regress



A deep CNN that learns statistical mean of input



No high frequency details

What is better: learn a distribution of real human faces

Find one sample x that satisfies: LR(x) = input

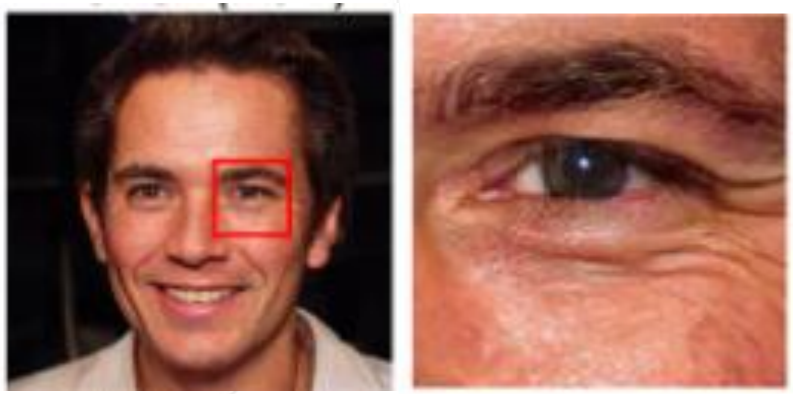
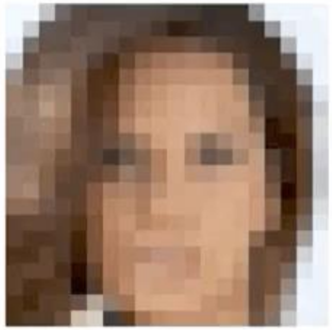


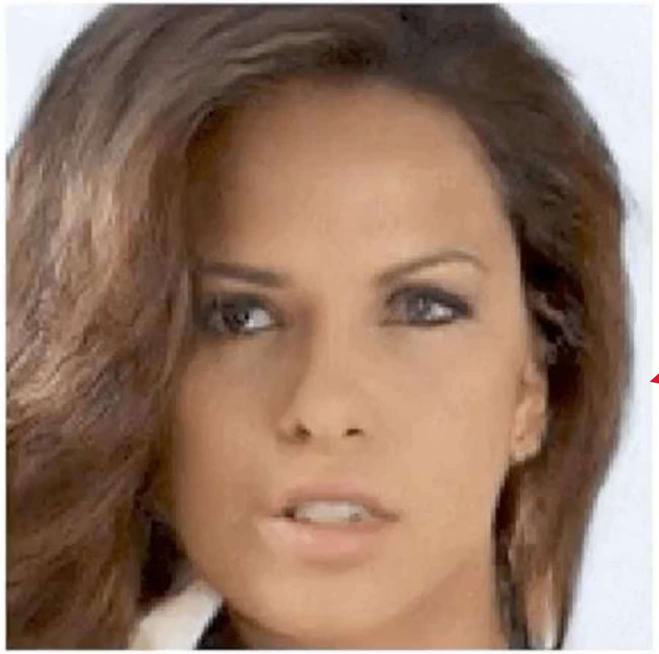
Image source: http://pulse.cs.duke.edu/, https://laptrinhx.com/how-to-train-stylegan-to-generate-realistic-faces-1888828401/

Introduction to Generative Image Models

■ Natural Image Distribution: The Ideal for Image Restoration

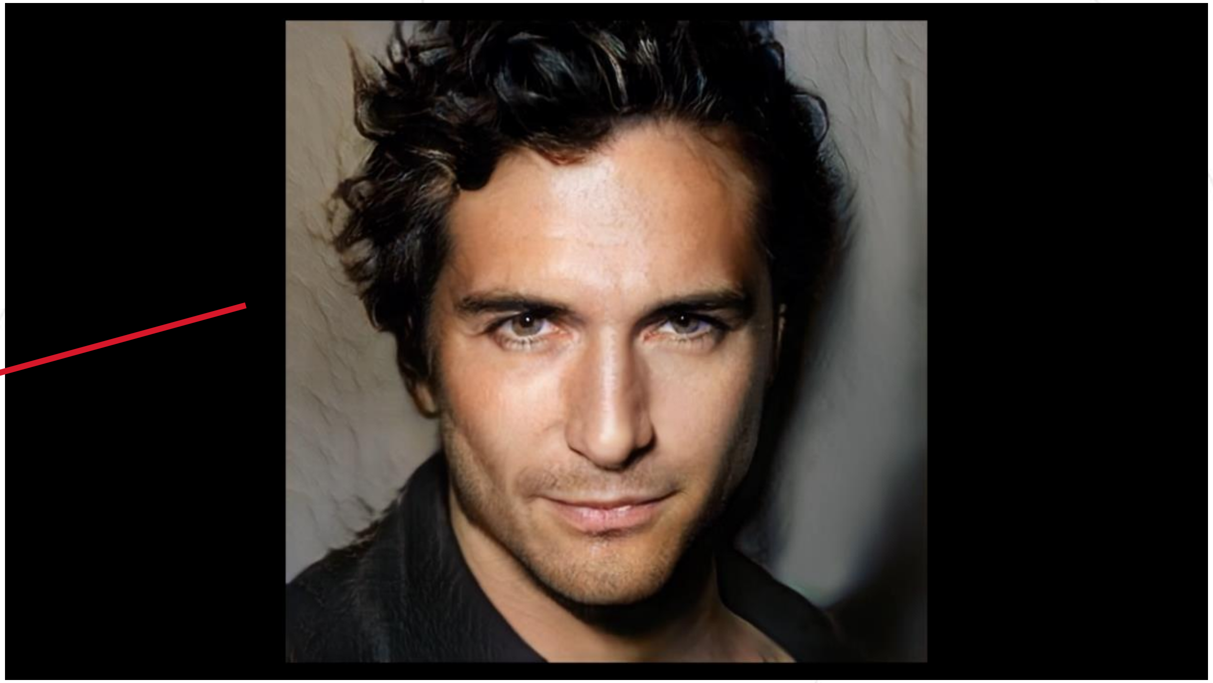


Corrupted
human face



Ideal reconstructions
(a conditional distribution)

Image source: Lugmayr et al, ECCV 2020: http://de.arxiv.org/pdf/2006.14200/

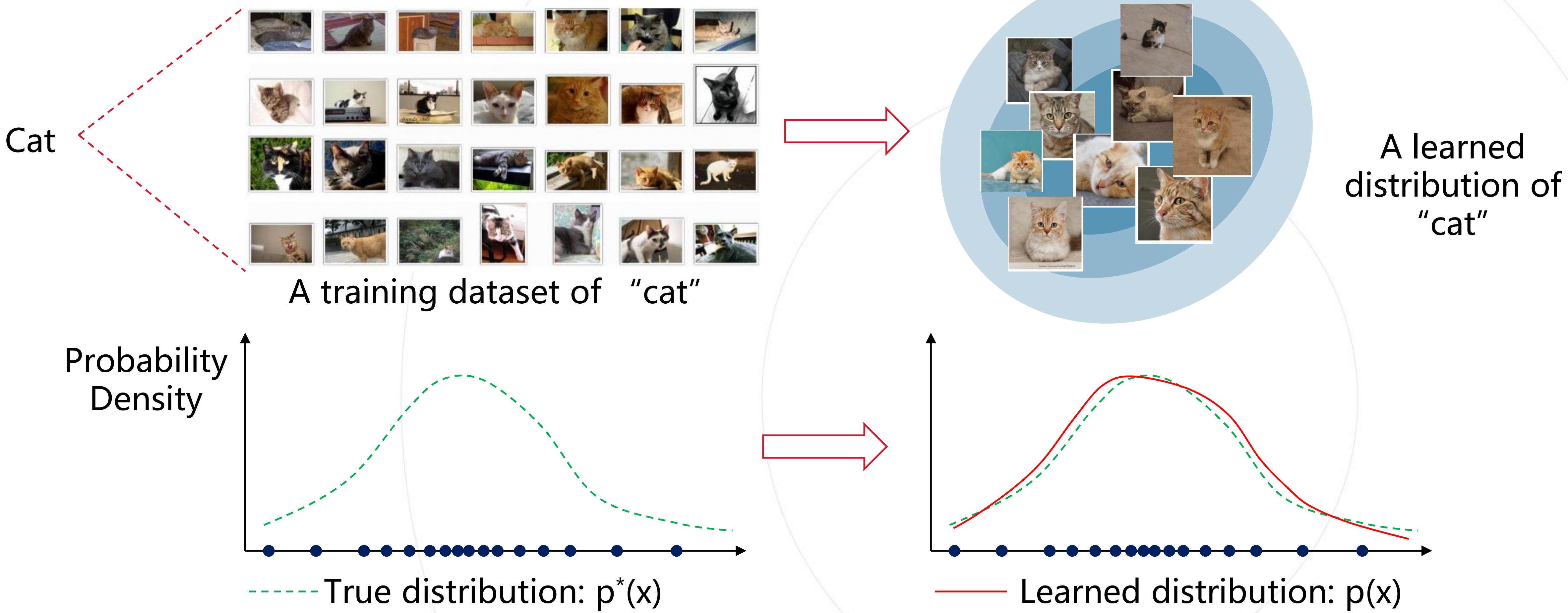


Natural image distribution of faces
(the marginal distribution)

Video source: https://www.youtube.com/watch?v=AnlUiFMD5lw

Introduction to Generative Image Models

- Generative Image Models: Learning the Distribution of Natural Images



Cat

A training dataset of “cat”

A learned distribution of “cat”

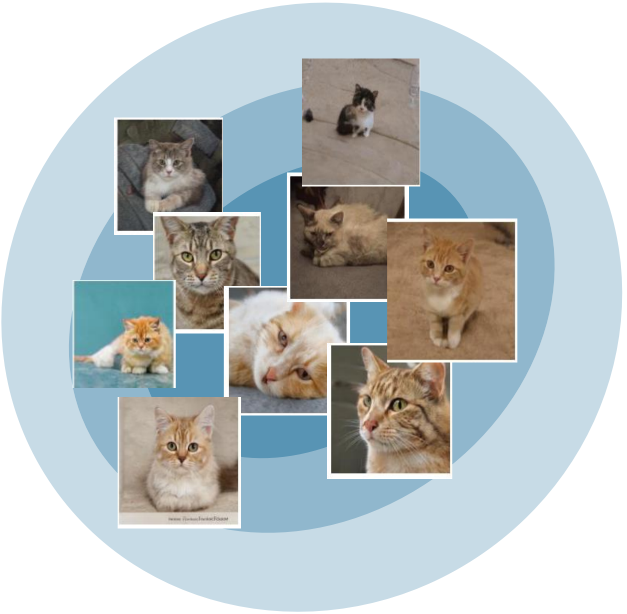
- - - - - True distribution: $p^*(x)$

—— Learned distribution: $p(x)$

Image source: http://thesecatsdonotexist.com/

Introduction to Generative Image Models

- Generative Image Models: Learning the Distribution of Natural Images



A learned distribution p(x)

Sampling from the learned distribution: x~p(x)

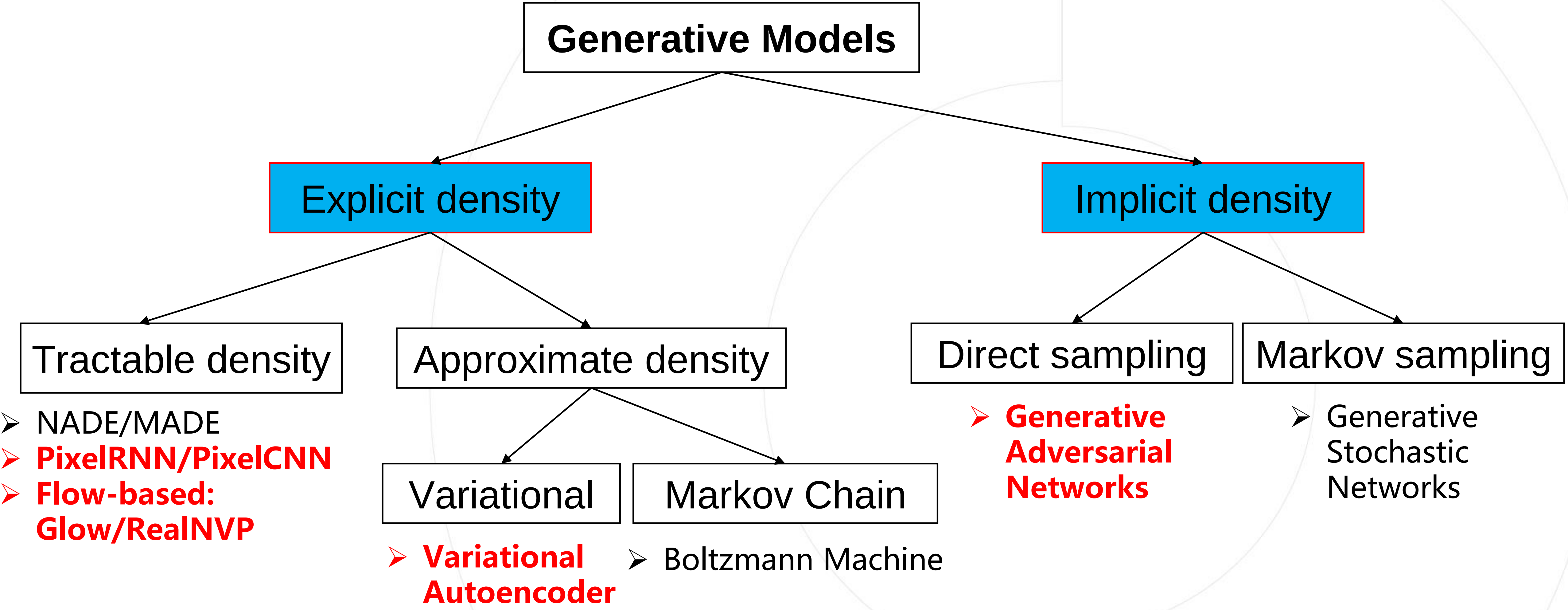
Generated samples

Generative models allow us to create new data that does not exist.

Image source: http://thesecatsdonotexist.com/

Introduction to Generative Image Models

- Taxnonomy of Generative Models



Generative Models

Explicit density

Implicit density

Tractable density

- NADE/MADE
- **PixelRNN/PixelCNN**
- **Flow-based: Glow/RealNVP**

Approximate density

Variational

- **Variational Autoencoder**

Markov Chain

- Boltzmann Machine

Direct sampling

- **Generative Adversarial Networks**

Markov sampling

- Generative Stochastic Networks

商汤 sensetime

清華大學 Tsinghua University

Generative Image Models

Part 1 Why Generative Image Models

Part 2 Explicit Density Image Models

Part 3 Generative Adversarial Networks

Part 4 Generative Image Models Beyond GANs

Explicit Density Models

- Explicit Density Models learns the density $P(\mathbf{x})$ of image pixels explicitly

For a low resolution image (e.g. 128*128), we need to model the joint distribution of 16k variables!
Modeling the joint dependency over massive variables is difficult.

Explicit Density Models

- Explicit Density Models learns the density $P(\mathbf{x})$ of image pixels explicitly

For a low resolution image (e.g. 128*128), we need to model the joint distribution of 16k variables! Modeling the joint dependency over massive variables is difficult.

- PixelRNN: Assume decomposable likelihood using chain rule

$$\underbrace{P(\mathbf{x})}_{\text{Likelihood of image}} = \prod_{i=1}^{n} \underbrace{P(x_i|x_1, x_2, \ldots, x_{i-1})}_{\text{Probability of } i\text{th pixel value given previously generated pixels}}$$

(which we model with a neural network!)

Explicit Density Models

- Explicit Density Models learns the density $P(\mathbf{x})$ of image pixels explicitly

For a low resolution image (e.g. 128*128), we need to model the joint distribution of 16k variables! Modeling the joint dependency over massive variables is difficult.

- PixelRNN: Assume decomposable likelihood using chain rule

$$\underbrace{P(\mathbf{x})} = \prod_{i=1}^{n} \underbrace{P(x_i|x_1, x_2, \dots, x_{i-1})}$$

Likelihood of image

Probability of ith pixel value given previously generated pixels

(which we model with a neural network!)

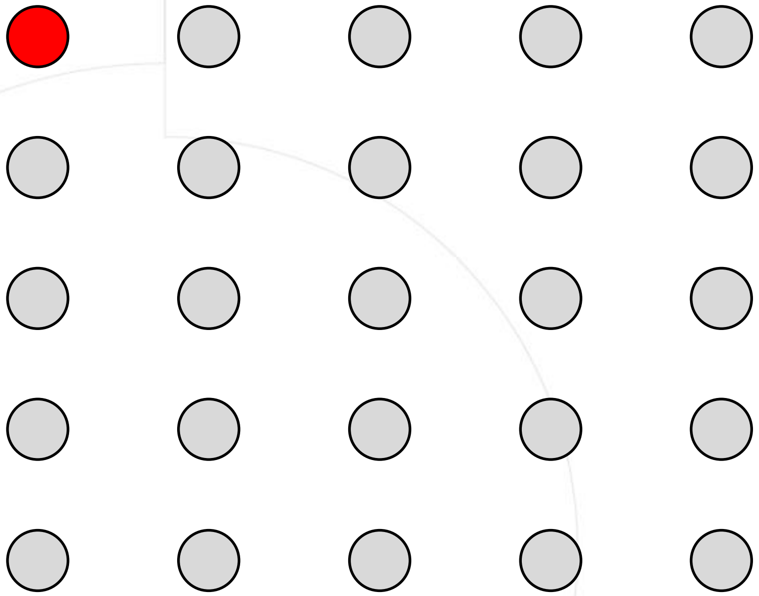
Testing: generate an image via $x_0 \sim P(x_0)$, $x_1 \sim P(x_1|x_0)$, $x_2 \sim P(x_2|x_1, x_0)$, etc.

Training: maximize the joint likelihood of a image dataset: $max_\theta \prod_{i=1}^{N} P(\mathbf{x}_i)$

Explicit Density Models

- PixelRNN [van der Oord et al. 2016]

Generate image pixels staring from the corner

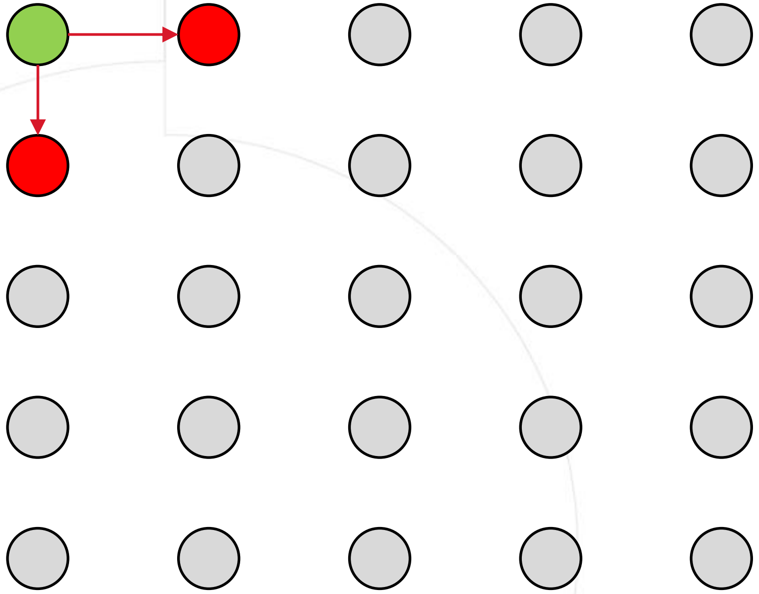


Explicit Density Models

- PixelRNN [van der Oord et al. 2016]

Generate image pixels staring from the corner

And proceed to the nearby pixels



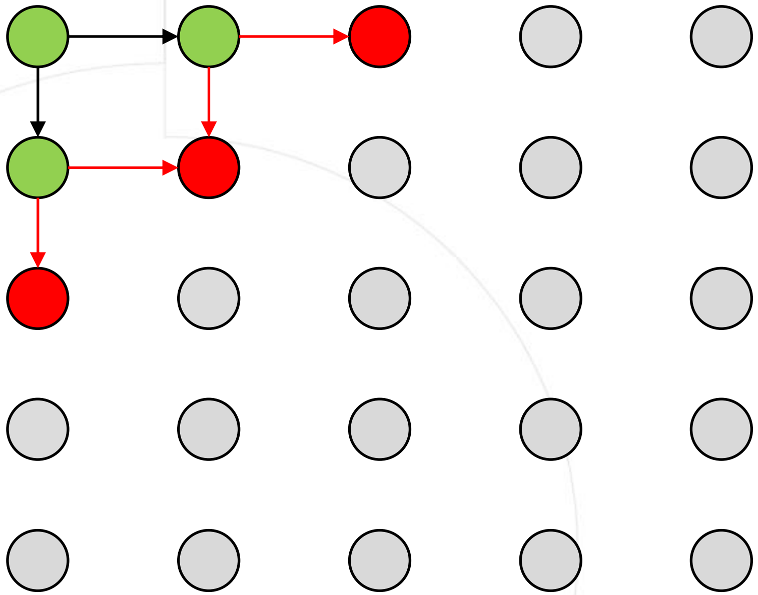
Explicit Density Models

- PixelRNN [van der Oord et al. 2016]

Generate image pixels staring from the corner

And proceed to the nearby pixels

Generating the current pixel is dependent on the triangle context from top-left region (RNNs like LSTM/GRUs are adopted to memorize this context)



Explicit Density Models

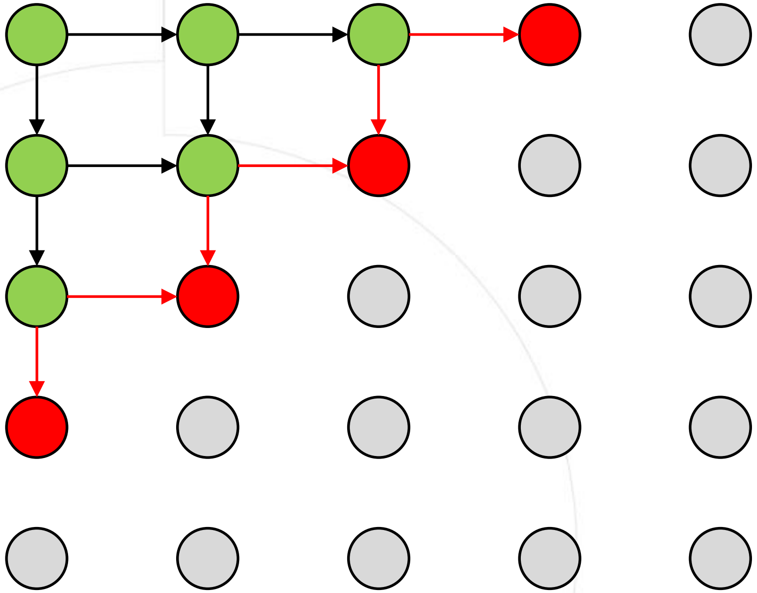
- PixelRNN [van der Oord et al. 2016]

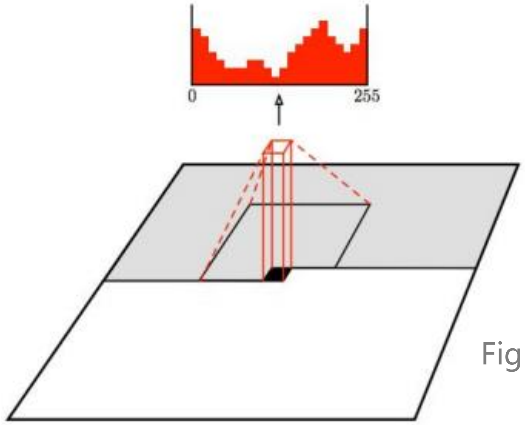
Generate image pixels staring from the corner

And proceed to the nearby pixels

Generating the current pixel is dependent on the triangle context from top-left region (RNNs like LSTM/GRUs are adopted to memorize this context)

For each pixel, output the softmax probability of pixels value over 0~255





More details referred to extended reading materials

$$P(x_i|x_1, x_2, \dots, x_{i-1}) \sim \text{Softmax}(0, 1, 2, \dots, 255)$$

Figure copyright van der Oord et al. 2016

Explicit Density Models

Pros and Cons of PixelRNN

Pros

- Very intuitive
- Calculation of $P(\mathbf{x})$ is tractable
- Good sample quality

Cons

- Very slow training/inference, generation must be sequential (e.g. taking 5 days on 8 GTX Titans to be trained on 32*32 CIFAR dataset)
- Cannot learn long-range context well

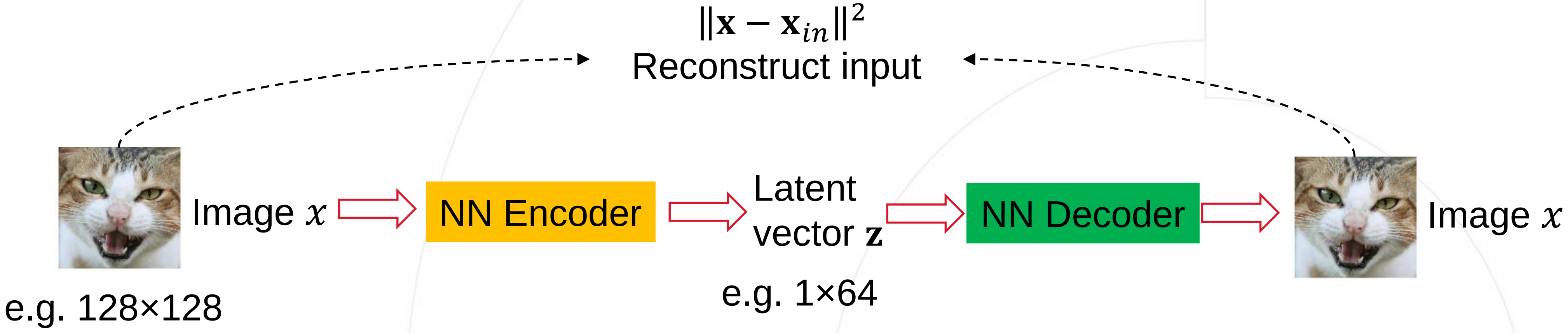
Generated 32*32 images by training on ImageNet

Figure copyright van der Oord et al. 2016

Explicit Density Models

■ Variational Autoencoder (VAE)

Some background: Autoencoder



$\|\mathbf{x} - \mathbf{x}_{in}\|^2$

Reconstruct input

Image x ⇨ NN Encoder ⇨ Latent vector $\mathbf{z}$ ⇨ NN Decoder ⇨ Image x

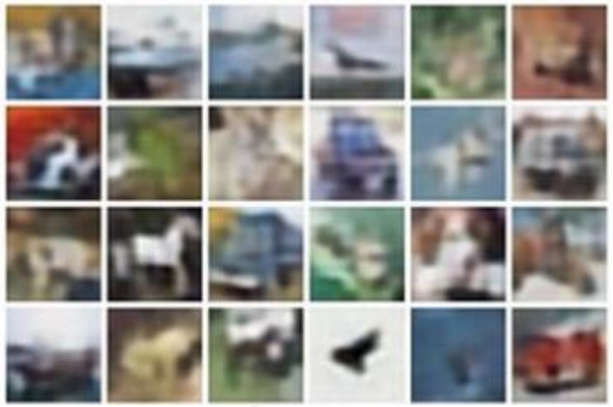
e.g. 128×128

e.g. 1×64

Input data

Encoder: 4-layer conv.

Decoder: 4-layer upconv.



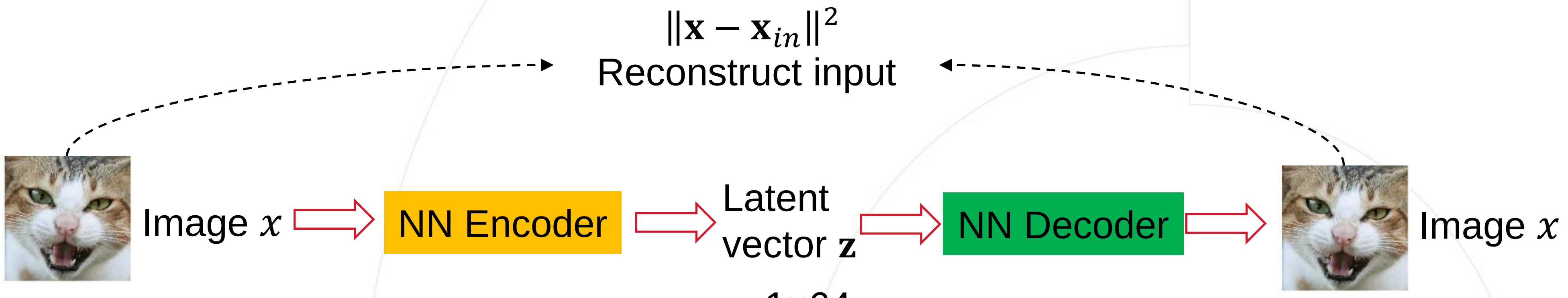
Reconstructed data

Image source: http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture11.pdf

Explicit Density Models

■ Variational Autoencoder (VAE)

From AE to VAE



$\|\mathbf{x} - \mathbf{x}_{in}\|^2$

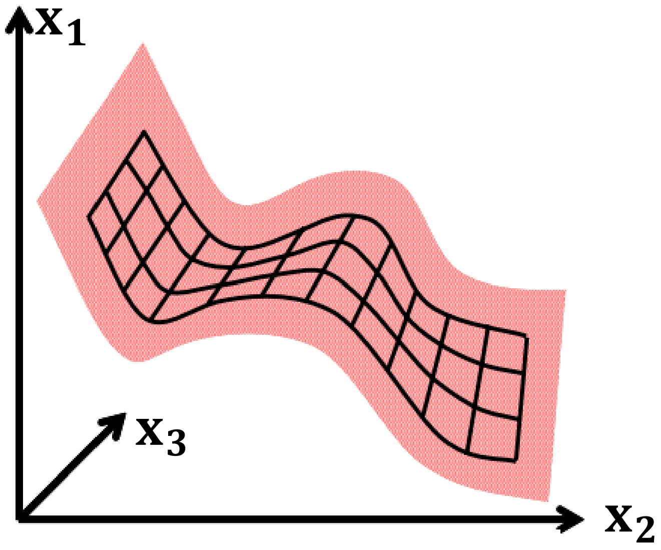
Reconstruct input

Image x ⇨ NN Encoder ⇨ Latent vector $\mathbf{z}$ ⇨ NN Decoder ⇨ Image x

e.g. 128×128

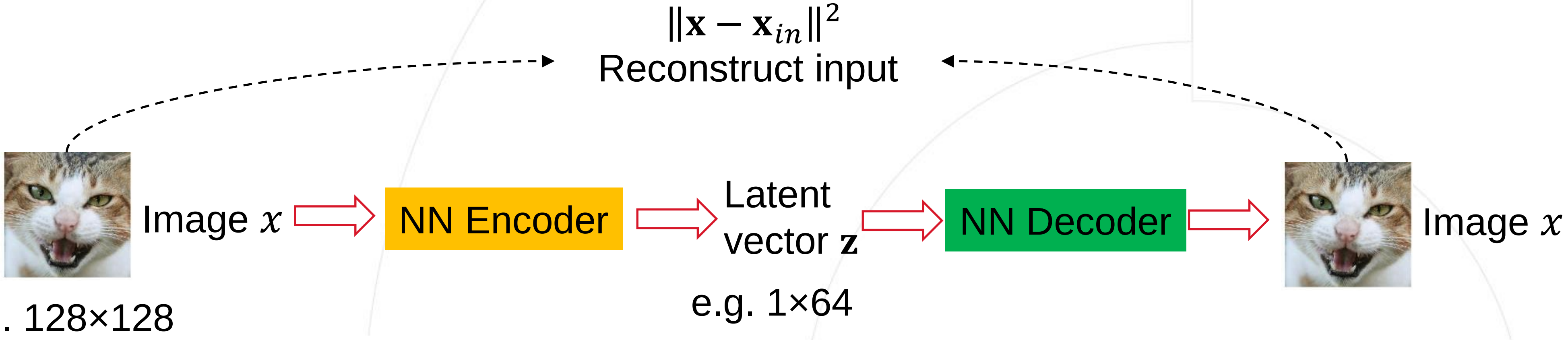
e.g. 1×64

The mapped observation $\mathbf{x}$ is high-dimensional, lying in a complex, uninterpretable distribution



■ Variational Autoencoder (VAE)

From AE to VAE



$\|\mathbf{x}-\mathbf{x}_{in}\|^2$

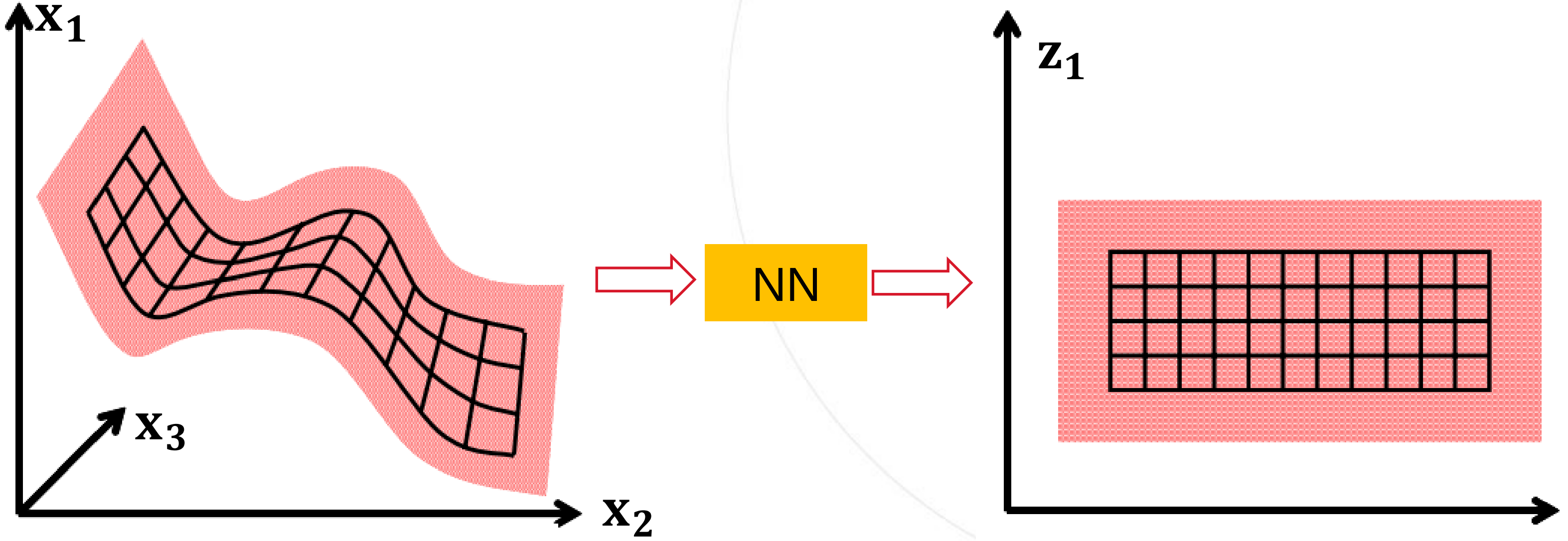
Reconstruct input

Image x ⇨ NN Encoder ⇨ Latent vector $\mathbf{z}$ ⇨ NN Decoder ⇨ Image x

e.g. 128×128

e.g. 1×64

The mapped observation $\mathbf{x}$ is high-dimensional, lying in a complex, uninterpretable distribution

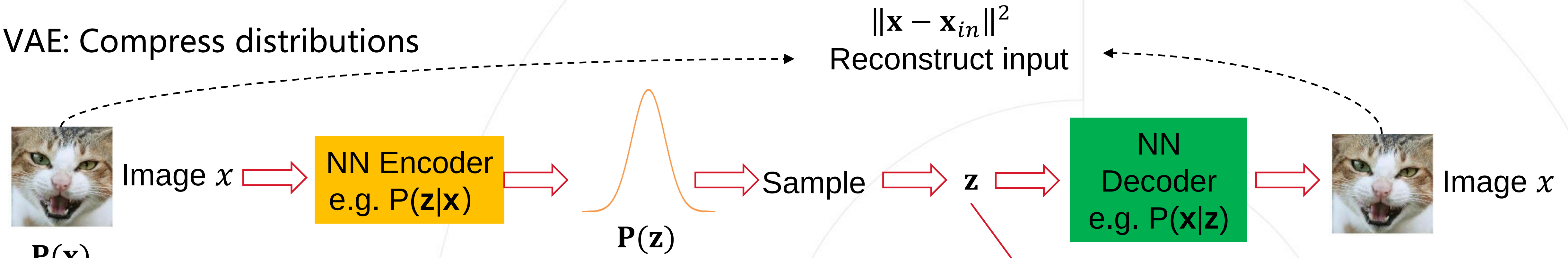


If the latent distribution $\mathbf{z}$ is low-dimensional, and simple, then everything will be easier!

Explicit Density Models

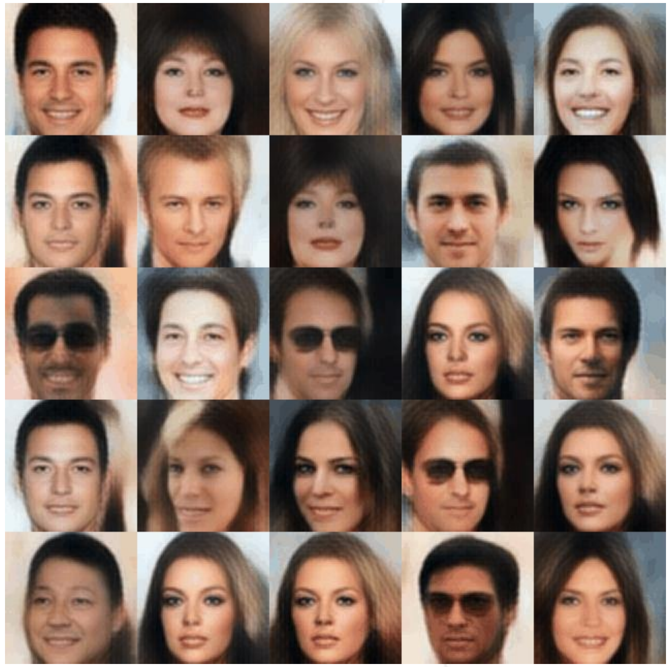
- Variational Autoencoder (VAE)

VAE: Compress distributions



$\mathbf{P}(\mathbf{x})$

The prior of latent vector z is assumed simple (e.g. multivariate Gaussian), which allows efficient sampling of P(z) instead of the more complicated P(x)



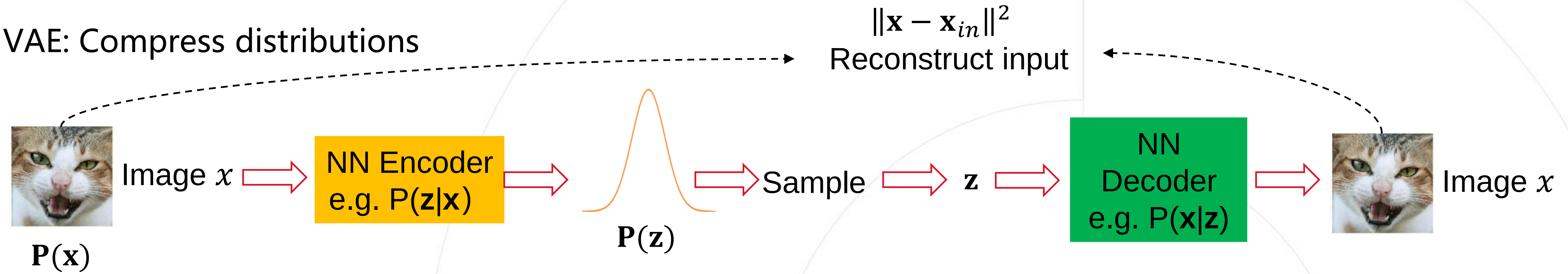
Sampled images from VAE

Image source: https://medium.com/vitrox-publication/generative-modeling-with-variational-auto-encoder-vae-fc449be9890e

Explicit Density Models

Variational Autoencoder (VAE)

VAE: Compress distributions



VAE optimizes intractable maximum likelihood, which is approximated by a tractable lower bound

$$max_{\theta}\prod_{i=1}^{N}P(\mathbf{x}_i),\quad \text{where}\quad P(\mathbf{x})=\int P(\mathbf{z})P(\mathbf{x}|\mathbf{z})\,d\mathbf{z}$$

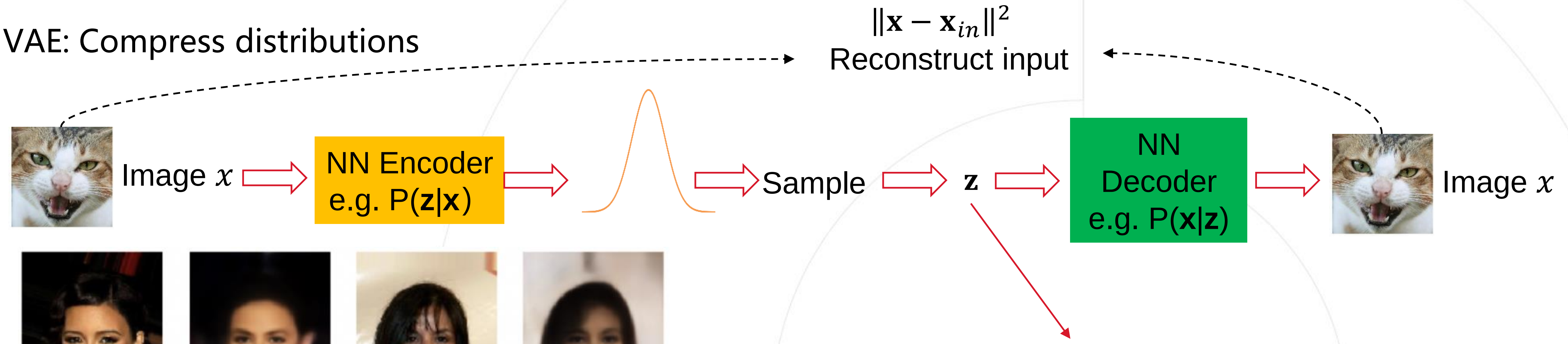
This is a simple prior

This conditional is modeled with NN

Explicit Density Models

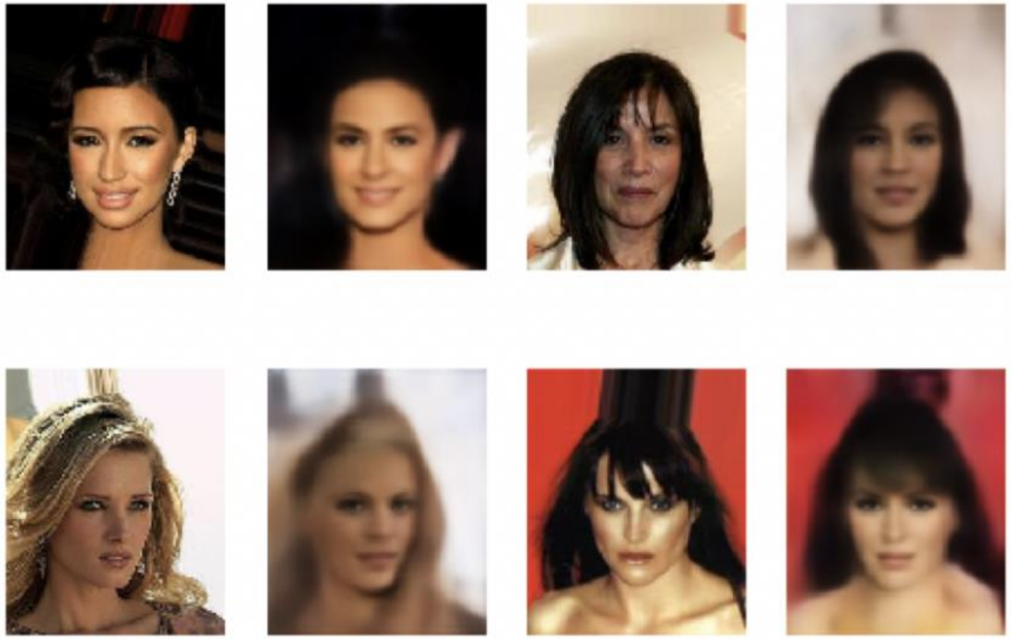
■ Variational Autoencoder (VAE)

VAE: Compress distributions



The prior of latent vector z is assumed simple (e.g. multivariate Gaussian), which allows efficient sampling of P(z) instead of the more complicated P(x)

Yet the reconstruction loss is independent for each pixel, which we know blurs the results



Images and VAE reconstructed versions

Tsinghua University

Generative Image Models

Part 1 Why Generative Image Models

Part 2 Explicit Density Models

Part 3 Generative Adversarial Networks

Part 4 Generative Image Models Beyond GANs

Generative Adversarial Networks

- So far …

Explicit density models all make efforts on approximating the marginal density P(x).

PixelRNN	Variational Autoencoder
Good samples, but slow sampling	Fast sampling, but bad samples

Generative Adversarial Networks

- So far …

Explicit density models all make efforts on approximating the marginal density P(x).

PixelRNN	Variational Autoencoder
Good samples, but slow sampling	Fast sampling, but bad samples

The evil lies in modeling the complex marginal P(x). Do we really have to?

Generative Adversarial Networks

- So far …

Explicit density models all make efforts on approximating the marginal density P(x).

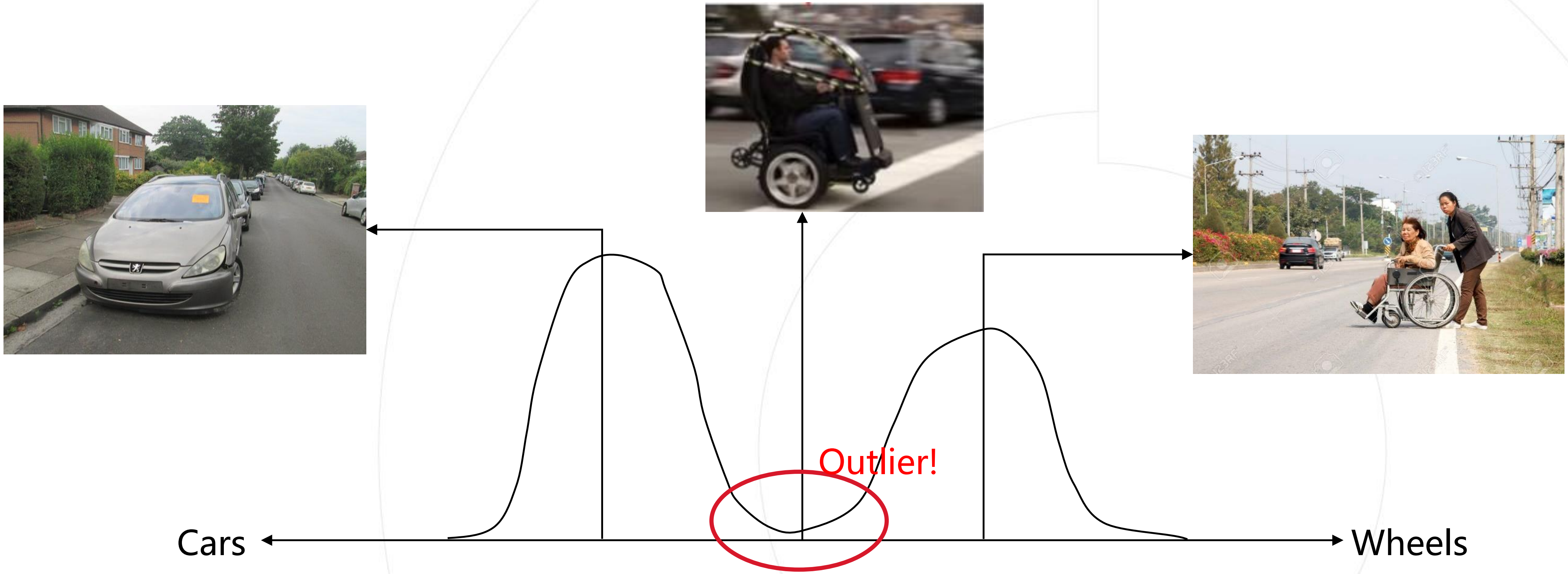
PixelRNN	Variational Autoencoder
Good samples, but slow sampling	Fast sampling, but bad samples

The evil lies in modeling the complex marginal P(x). Do we really have to?

The answer depends. If you build modes for outlier detection, yes

Generative Adversarial Networks

- Just an Episode: Modeling the Density P(x) for Outlier Detection



A learned distribution p(x)

Generative model detects outliers that do not match the training distribution. It is desired for perception systems (e.g. auto-driving) to detect anomalies before a wrong decision is made.

Generative Adversarial Networks

- So far …

Explicit density models all make efforts on approximating the marginal density P(x).

PixelRNN	Variational Autoencoder
Good samples, but slow sampling	Fast sampling, but bad samples

The evil lies in modeling the complex marginal P(x). Do we really have to?

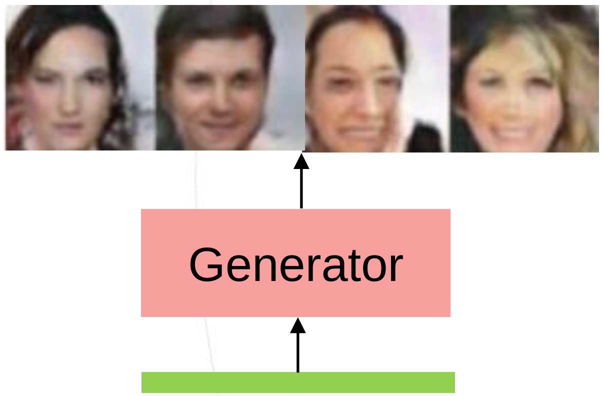
The answer depends. If you build modes for outlier detection, yes

However in most cases, we only care about sampling: x ~ P(x), not exactly modeling P(x)

Generative Adversarial Networks

■ Generative Adversarial Networks

In GANs, we learn a generator that reconstructs high fidelity samples from latent vectors (noise)

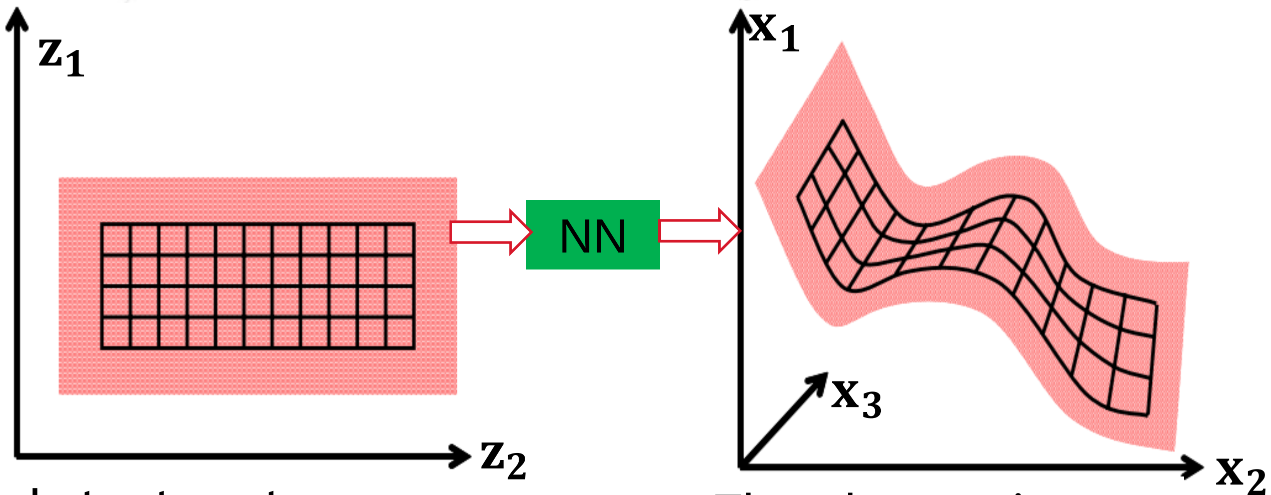


Fake images (from generator)

Generator

Latent vector z (also called noise, sampled from a simple prior)

Image source: Ward et al., 3D Surface Parameterization Using Manifold Learning for Medial Shape Representation, 2007



Latent vector space

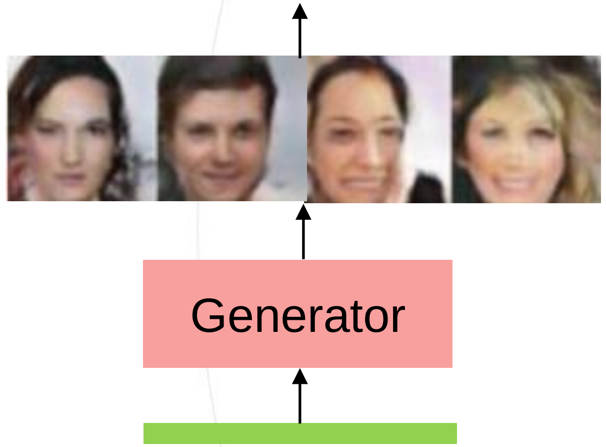
The observation space

Generative Adversarial Networks

■ Generative Adversarial Networks

In GANs, we learn a generator that reconstructs high fidelity samples from latent vectors (noise)

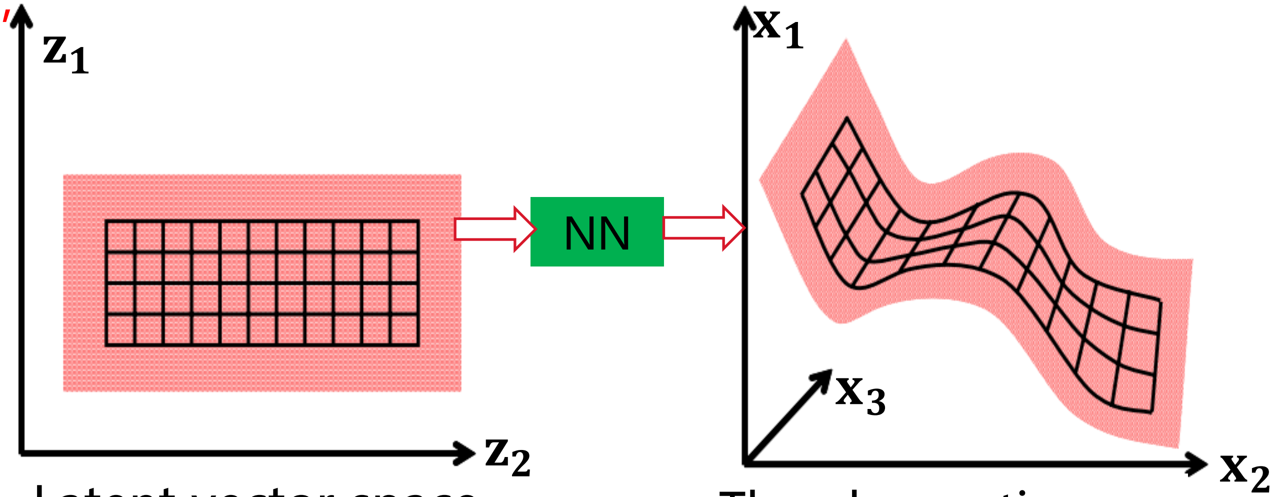
A ***learned*** loss function characterizing how good the generated samples are, providing training signals



Fake images (from generator)

Latent vector z (also called noise, sampled from a simple prior)

Image source: Ward et al., 3D Surface Parameterization Using Manifold Learning for Medial Shape Representation, 2007



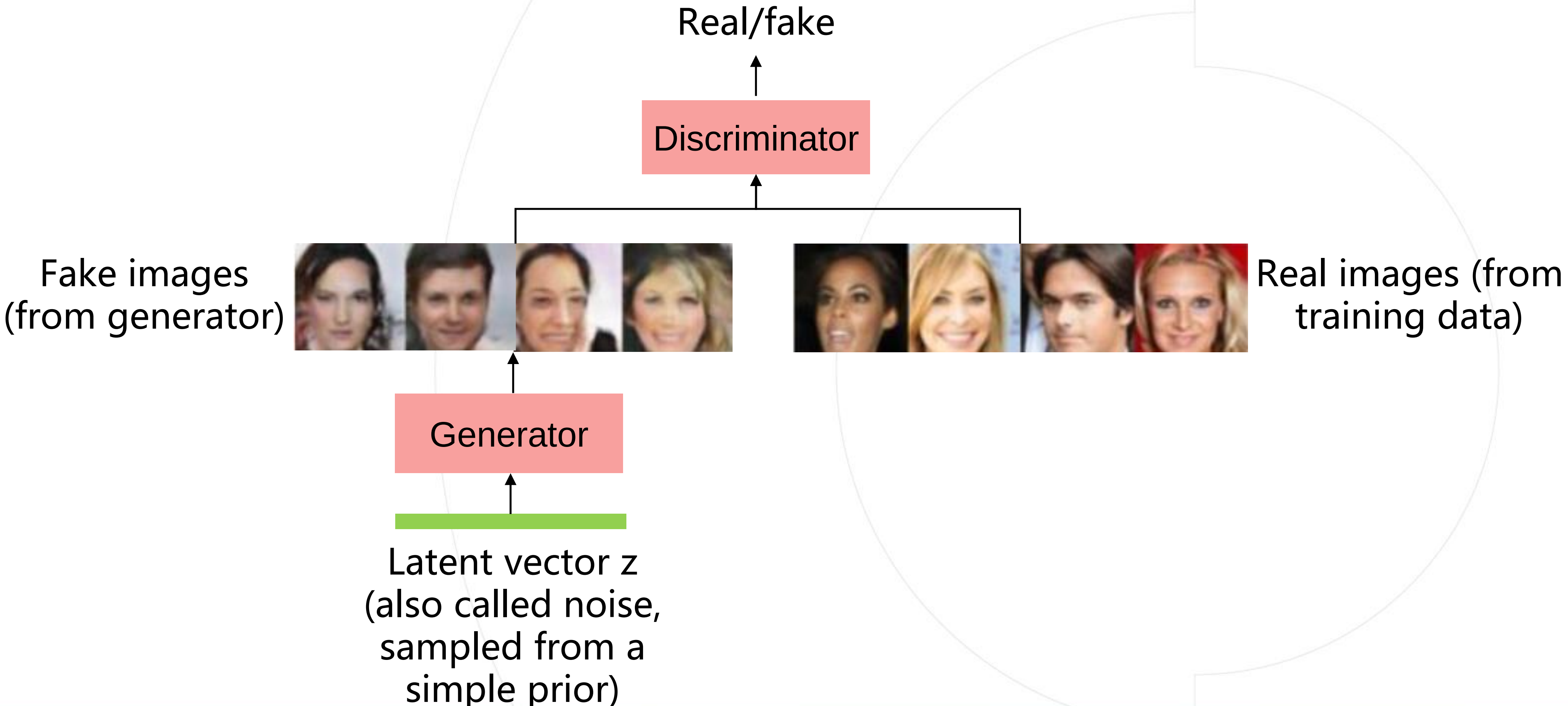
Latent vector space

The observation space

Generative Adversarial Networks

■ Generative Adversarial Networks

Learning a metric of the true density P(x) is difficult. Solving it is the most amazing part of GAN.



Real/fake

Discriminator

Fake images (from generator)

Real images (from training data)

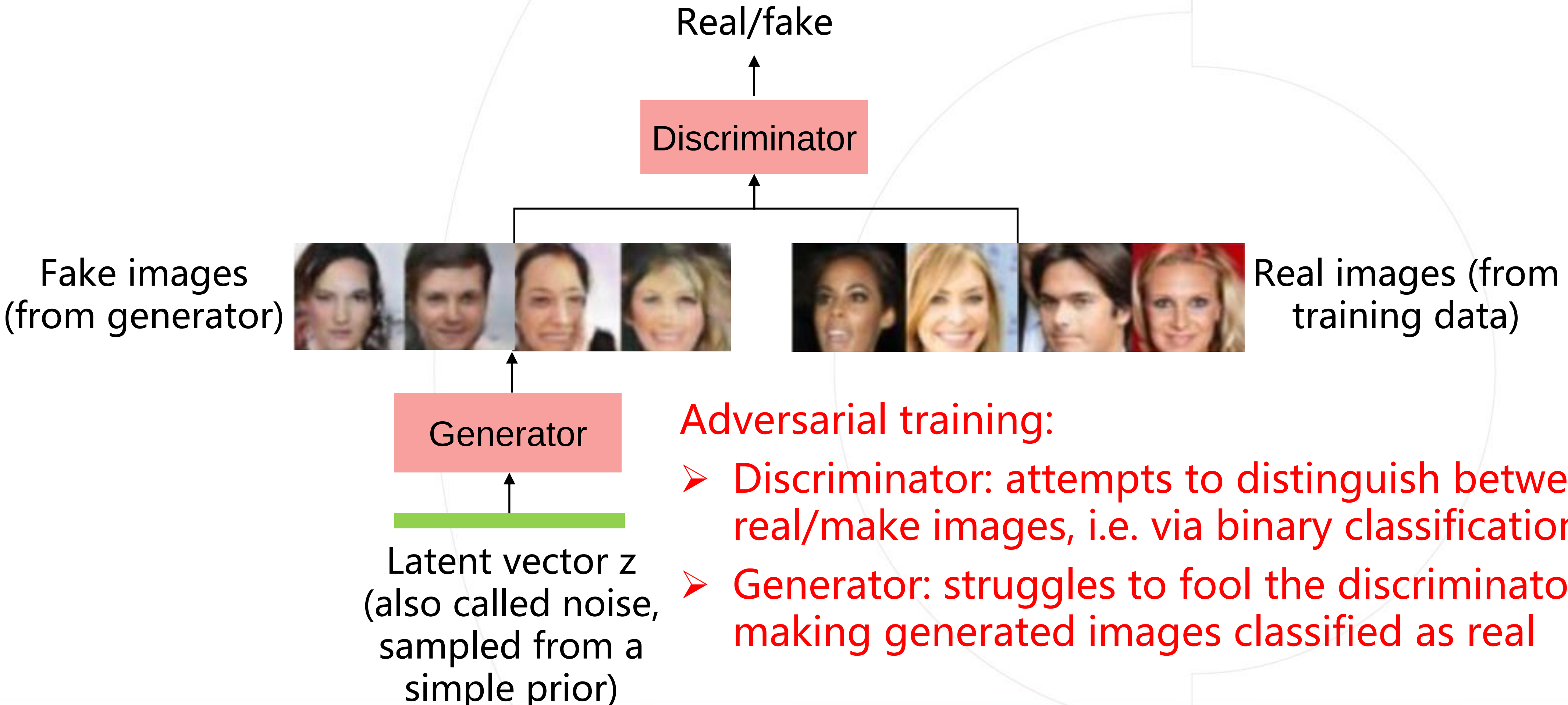
Generator

Latent vector z (also called noise, sampled from a simple prior)

Generative Adversarial Networks

Generative Adversarial Networks

Learning a metric of the true density P(x) is difficult. Solving it is the most amazing part of GAN.



Real/fake
Discriminator
Fake images (from generator)
Real images (from training data)
Generator
Latent vector z (also called noise, sampled from a simple prior)

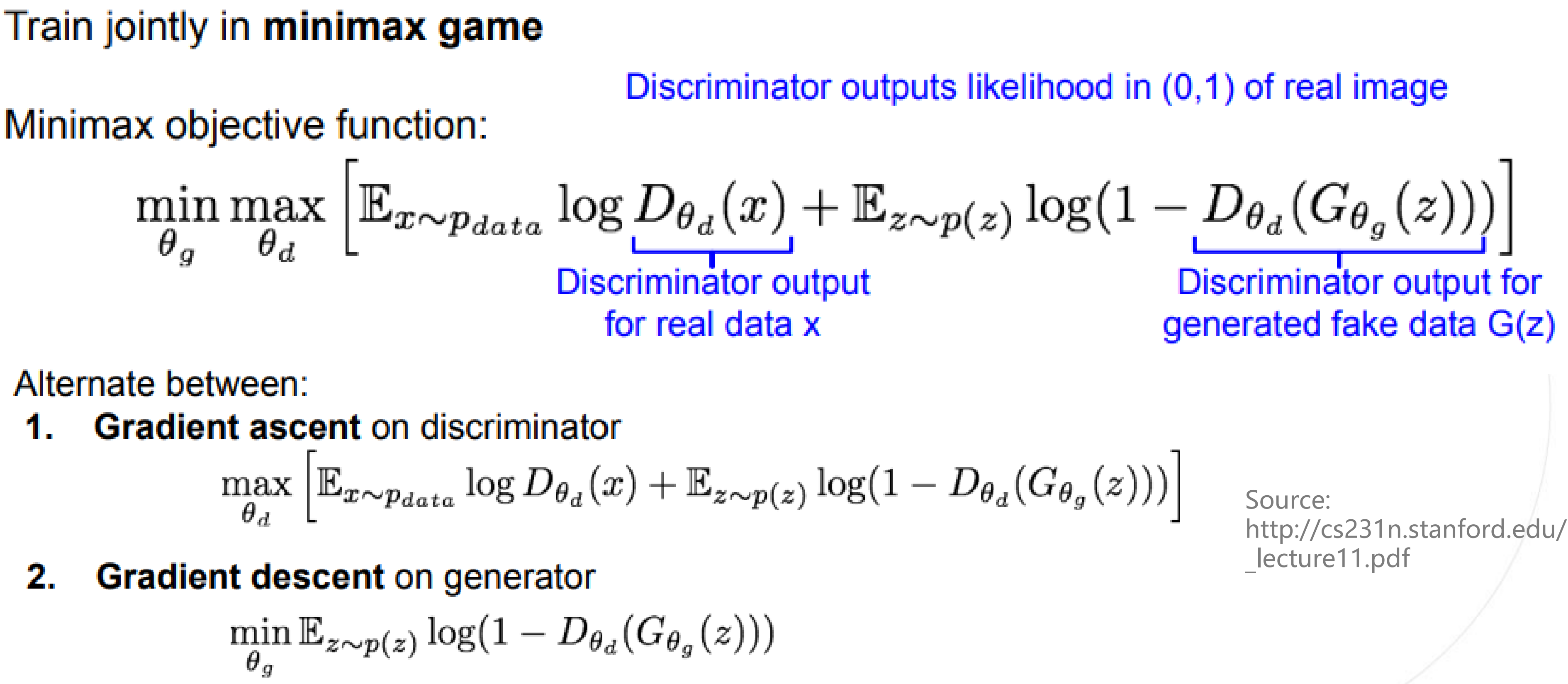
Adversarial training:

- Discriminator: attempts to distinguish between real/make images, i.e. via binary classification
- Generator: struggles to fool the discriminator, e.g. making generated images classified as real

Generative Adversarial Networks

Generative Adversarial Networks

Learning a metric of the true density P(x) is difficult. Solving it is the most amazing part of GAN.



Train jointly in **minimax game**

Minimax objective function:

Discriminator outputs likelihood in (0,1) of real image

$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data x}} + \mathbb{E}_{z \sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data G(z)}}) \right]$$

Alternate between:

1. **Gradient ascent** on discriminator

$$\max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log D_{\theta_d}(x) + \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z))) \right]$$

2. **Gradient descent** on generator

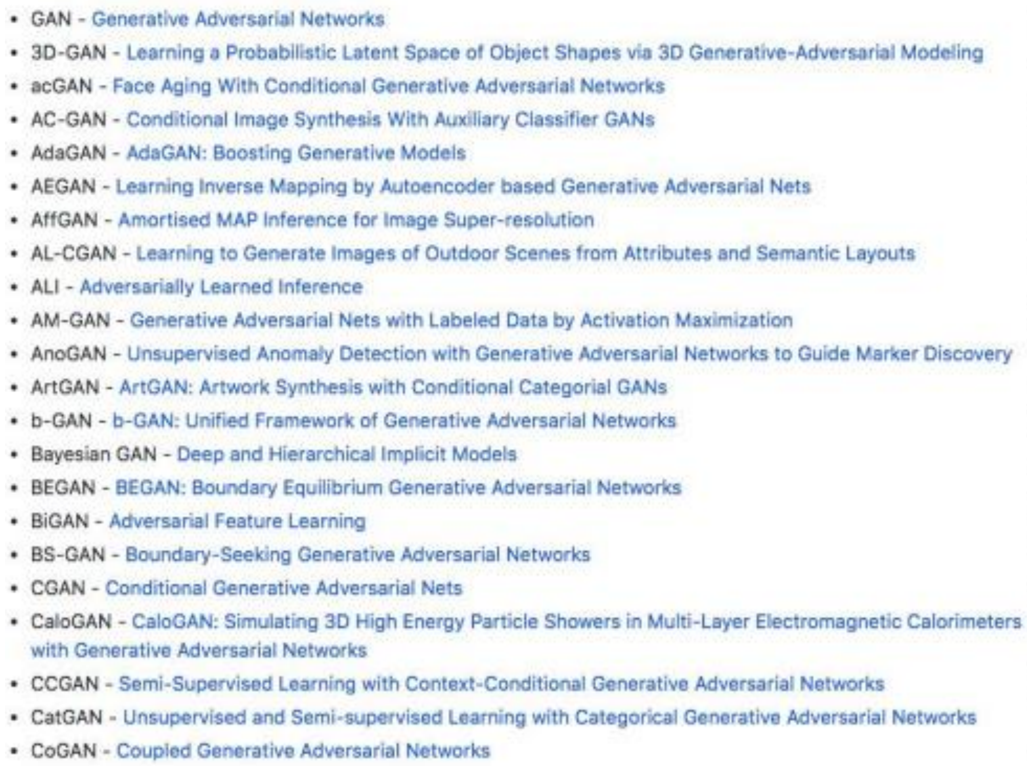
$$\min_{\theta_g} \mathbb{E}_{z \sim p(z)} \log(1 - D_{\theta_d}(G_{\theta_g}(z)))$$

Source:
http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture11.pdf

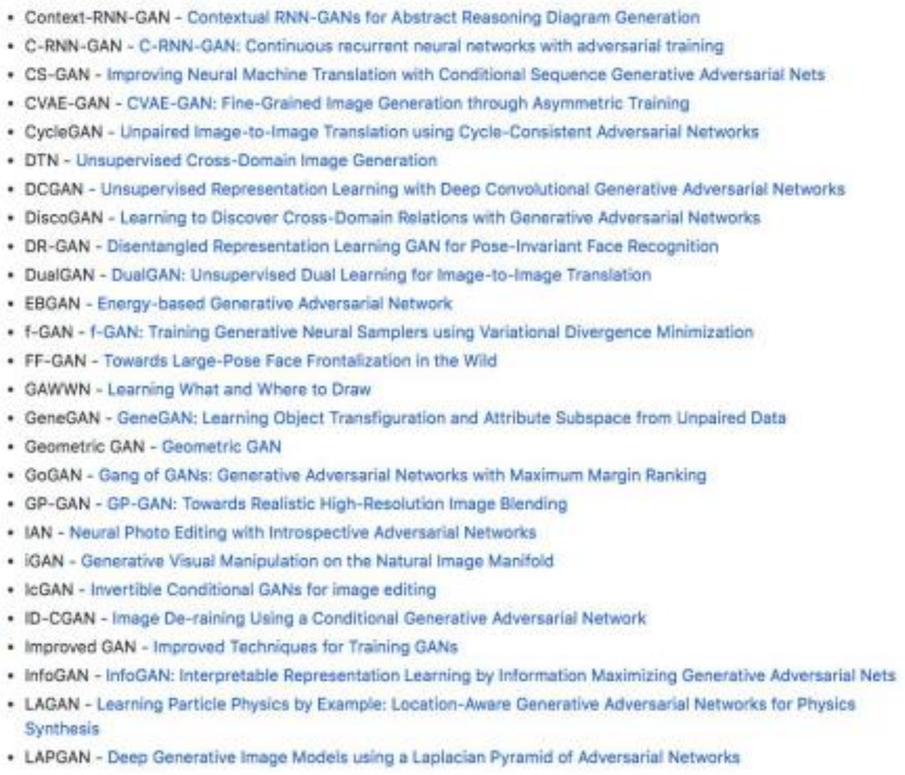
Generative Adversarial Networks

■ Why We Love GANs

https://github.com/hindupuravinash/the-gan-zoo



- GAN - Generative Adversarial Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- AdaGAN - AdaGAN: Boosting Generative Models
- AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts
- ALI - Adversarially Learned Inference
- AM-GAN - Generative Adversarial Nets with Labeled Data by Activation Maximization
- AnoGAN - Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery
- ArtGAN - ArtGAN: Artwork Synthesis with Conditional Categorial GANs
- b-GAN - b-GAN: Unified Framework of Generative Adversarial Networks
- Bayesian GAN - Deep and Hierarchical Implicit Models
- BEGAN - BEGAN: Boundary Equilibrium Generative Adversarial Networks
- BiGAN - Adversarial Feature Learning
- BS-GAN - Boundary-Seeking Generative Adversarial Networks
- CGAN - Conditional Generative Adversarial Nets
- CaloGAN - CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks
- CCGAN - Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks
- CatGAN - Unsupervised and Semi-supervised Learning with Categorical Generative Adversarial Networks
- CoGAN - Coupled Generative Adversarial Networks



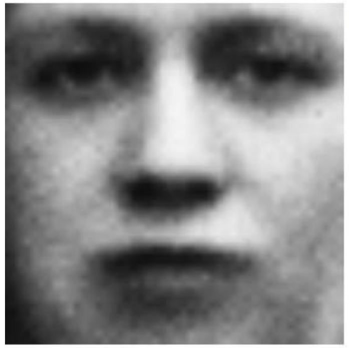
- Context-RNN-GAN - Contextual RNN-GANs for Abstract Reasoning Diagram Generation
- C-RNN-GAN - C-RNN-GAN: Continuous recurrent neural networks with adversarial training
- CS-GAN - Improving Neural Machine Translation with Conditional Sequence Generative Adversarial Nets
- CVAE-GAN - CVAE-GAN: Fine-Grained Image Generation through Asymmetric Training
- CycleGAN - Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks
- DTN - Unsupervised Cross-Domain Image Generation
- DCGAN - Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
- DiscoGAN - Learning to Discover Cross-Domain Relations with Generative Adversarial Networks
- DR-GAN - Disentangled Representation Learning GAN for Pose-Invariant Face Recognition
- DualGAN - DualGAN: Unsupervised Dual Learning for Image-to-Image Translation
- EBGAN - Energy-based Generative Adversarial Network
- f-GAN - f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization
- FF-GAN - Towards Large-Pose Face Frontalization in the Wild
- GAWWN - Learning What and Where to Draw
- GeneGAN - GeneGAN: Learning Object Transfiguration and Attribute Subspace from Unpaired Data
- Geometric GAN - Geometric GAN
- GoGAN - Gang of GANs: Generative Adversarial Networks with Maximum Margin Ranking
- GP-GAN - GP-GAN: Towards Realistic High-Resolution Image Blending
- IAN - Neural Photo Editing with Introspective Adversarial Networks
- iGAN - Generative Visual Manipulation on the Natural Image Manifold
- IcGAN - Invertible Conditional GANs for image editing
- ID-CGAN - Image De-raining Using a Conditional Generative Adversarial Network
- Improved GAN - Improved Techniques for Training GANs
- InfoGAN - InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
- LAGAN - Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis
- LAPGAN - Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks

- ➢ GAN is an active research area with astonishingly large family.
- ➢ GANs are applied to almost everywhere, e.g. computer vision, NLP, medical imaging, biocomputing, speech recognition, etc.
- ➢ Why we love GANs so much?

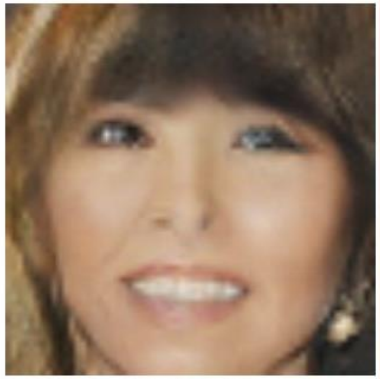
Generative Adversarial Networks

- Why We Love GANs: Really High Quality Sample Generation

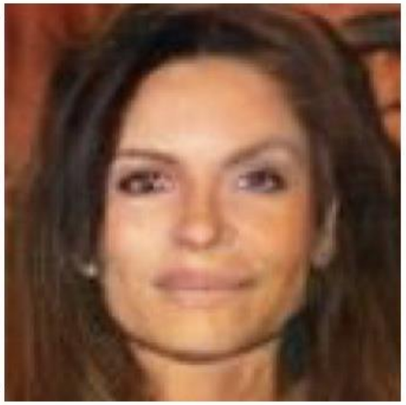
Ian Goodfellow on Twitter: "4 years of GAN progress"



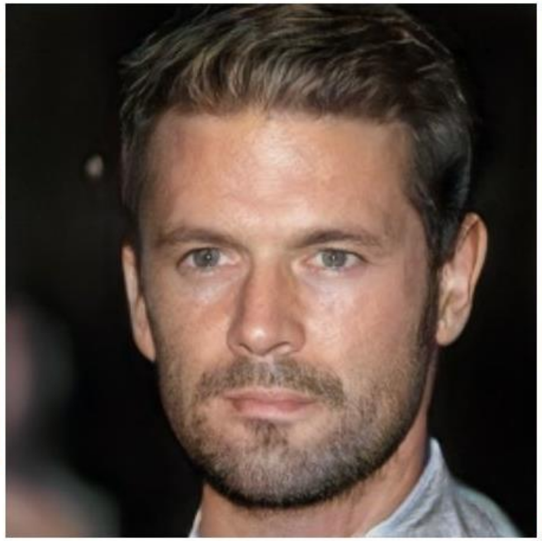
2014
The OG GAN



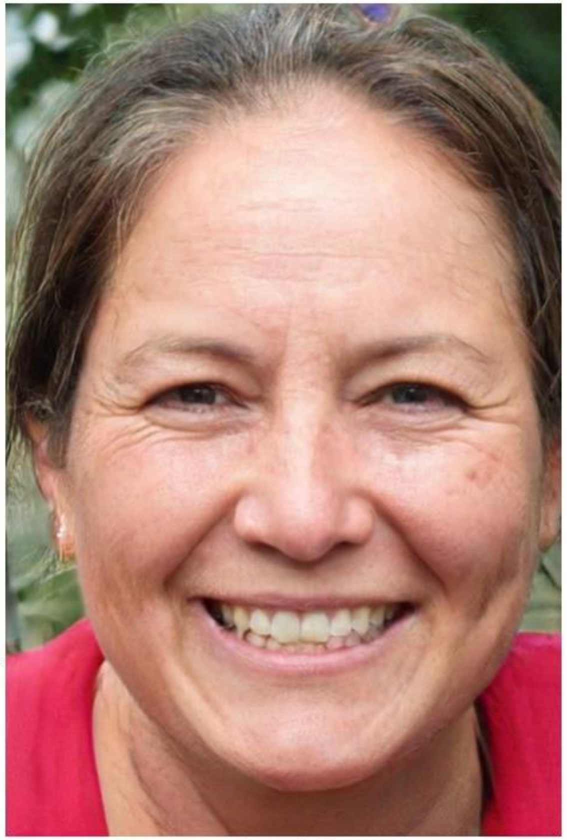
2015
DCGAN



2016
Coupled GAN



2017
Progressively Growing GAN

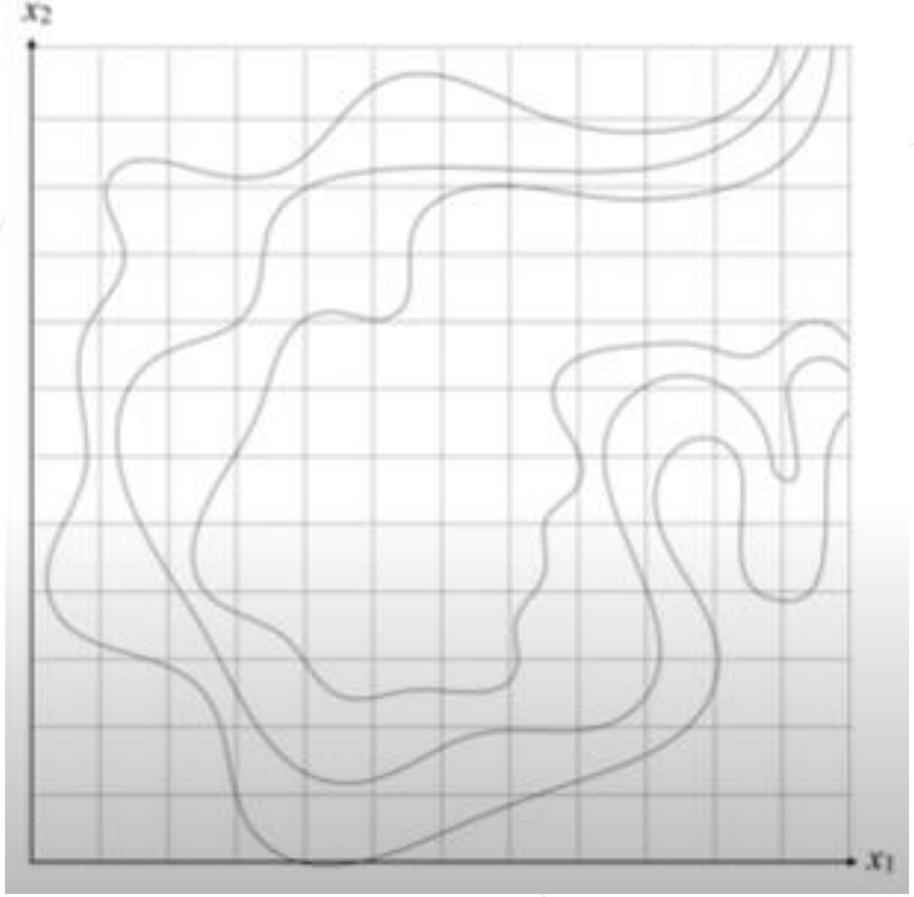


2018
StyleGAN

OG GAN: Goodfellow et al. Generative Adversarial Networks, 2014
DCGAN: Radford et al. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks, 2015
Coupled GAN: Ming-Yu et al. Coupled Generative Adversarial Networks, 2016
Progressively Growing GAN: Karras et al. Progressive Growing of GANs for Improved Quality, Stability, and Variation, 2017
StyleGAN: Karras et al. A Style-Based Generator Architecture for Generative Adversarial Networks, 2018

Why We Love GANs: Really High Quality Sample Generation

Why high quality samples?

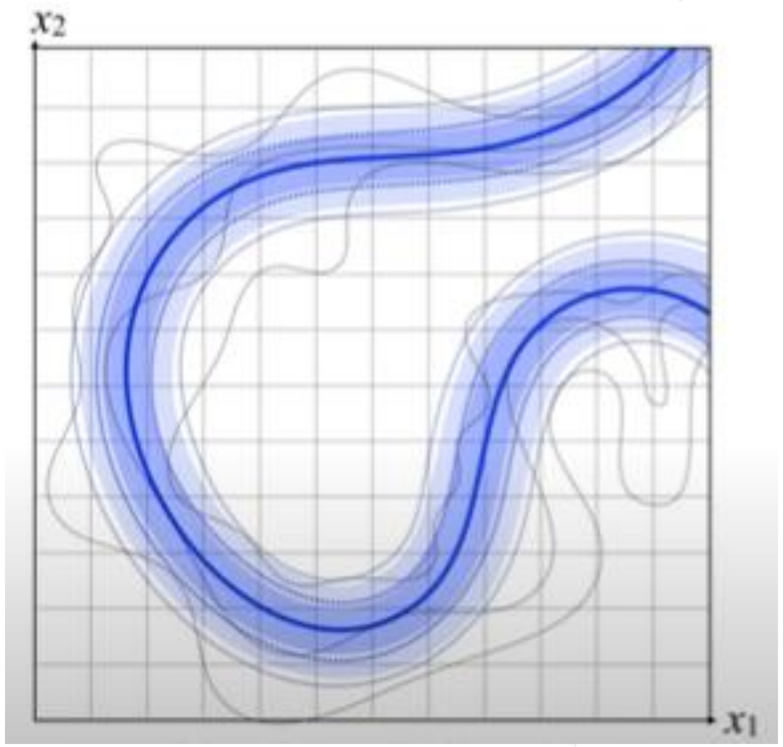


The natural image manifold, only a two-dim cartoon. Most space is empty (images do not look natural).

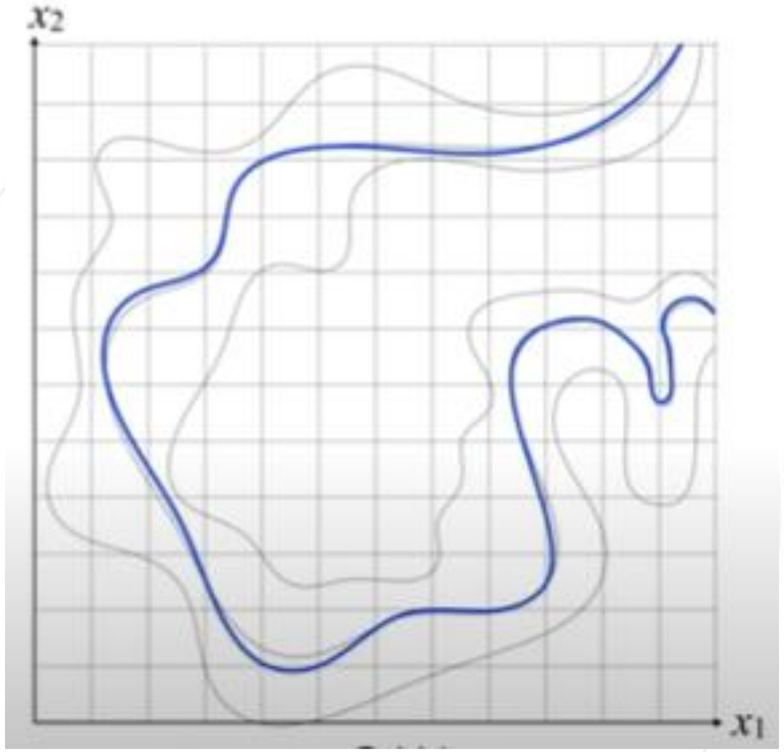
Image source: MIT 6.S191 (2018): Deep Generative Modeling

■ Why We Love GANs: Really High Quality Sample Generation

Why high quality samples?



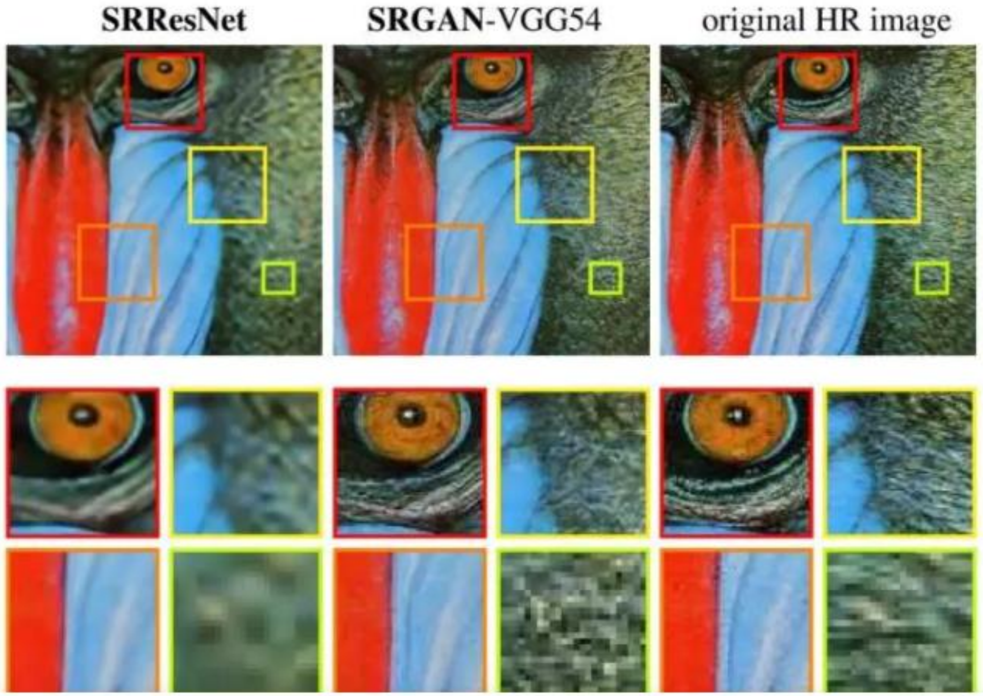
To satisfy joint likelihood objective the probability mass needs to spread over the manifold

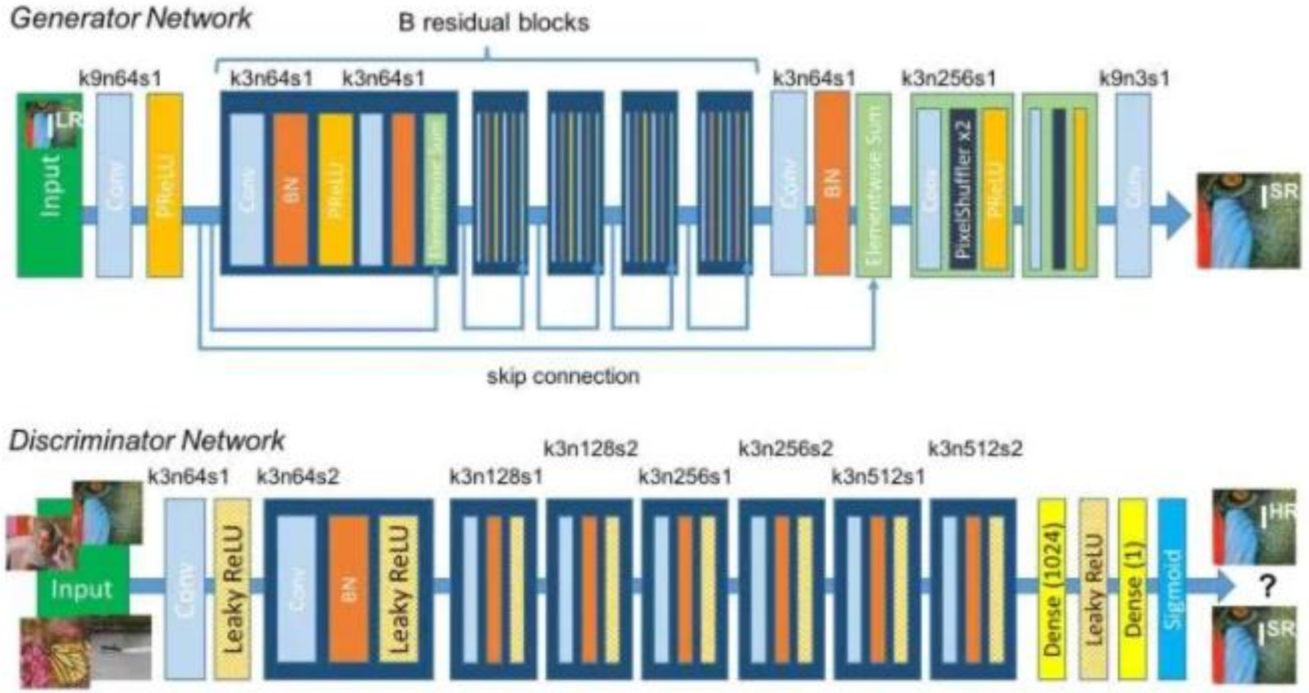


To satisfy GANs objective only need to model a portion of it instead of whole (but with high quality)

Image source: MIT 6.S191 (2018): Deep Generative Modeling

Example: SRGAN/ESRGAN





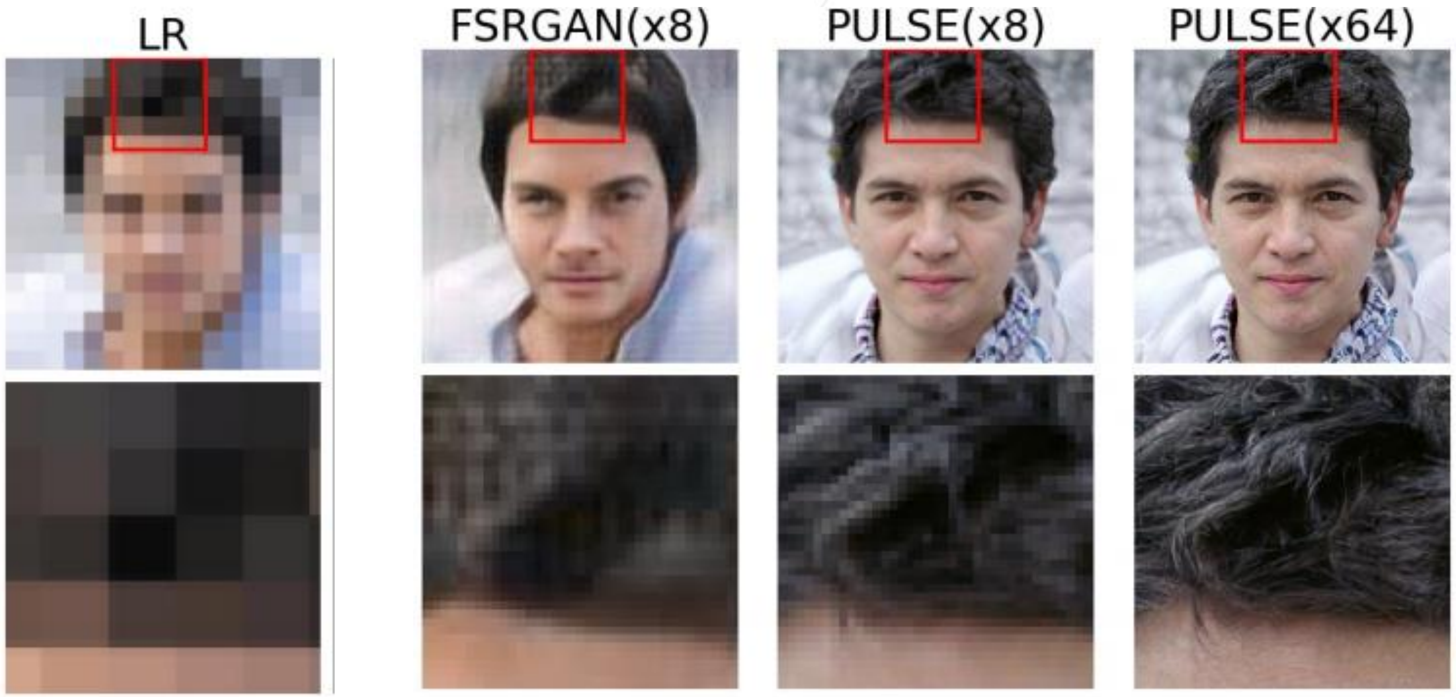
$$l_X^{SR} = l_{MSE}^{SR} + 10^{-6} * l_{Gen}^{SR}$$

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

SRGAN: https://arxiv.org/abs/1609.04802
ESRGAN: https://arxiv.org/abs/1809.00219

Generative Adversarial Networks

- Example: PULSE



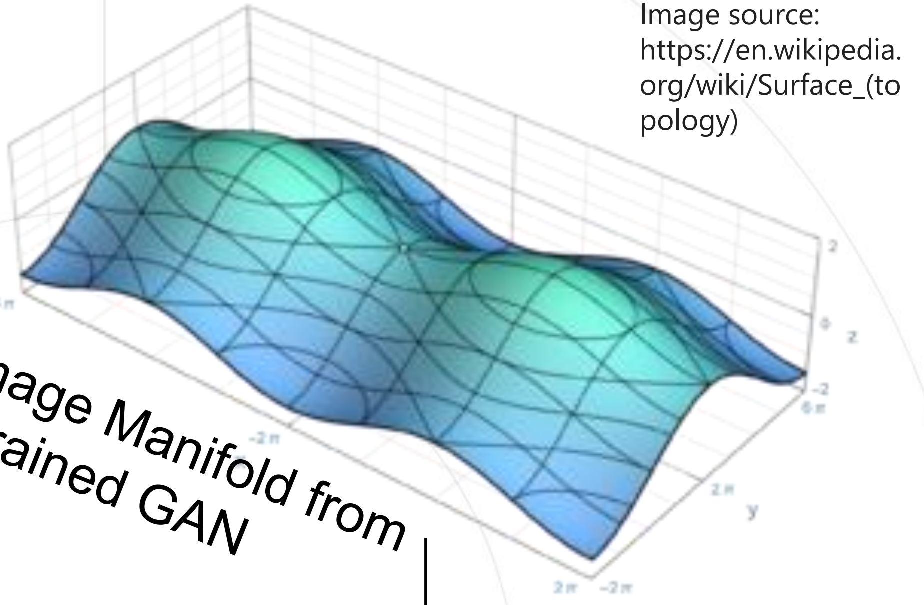


Image source: https://en.wikipedia.org/wiki/Surface_(topology)

Image Manifold from a trained GAN

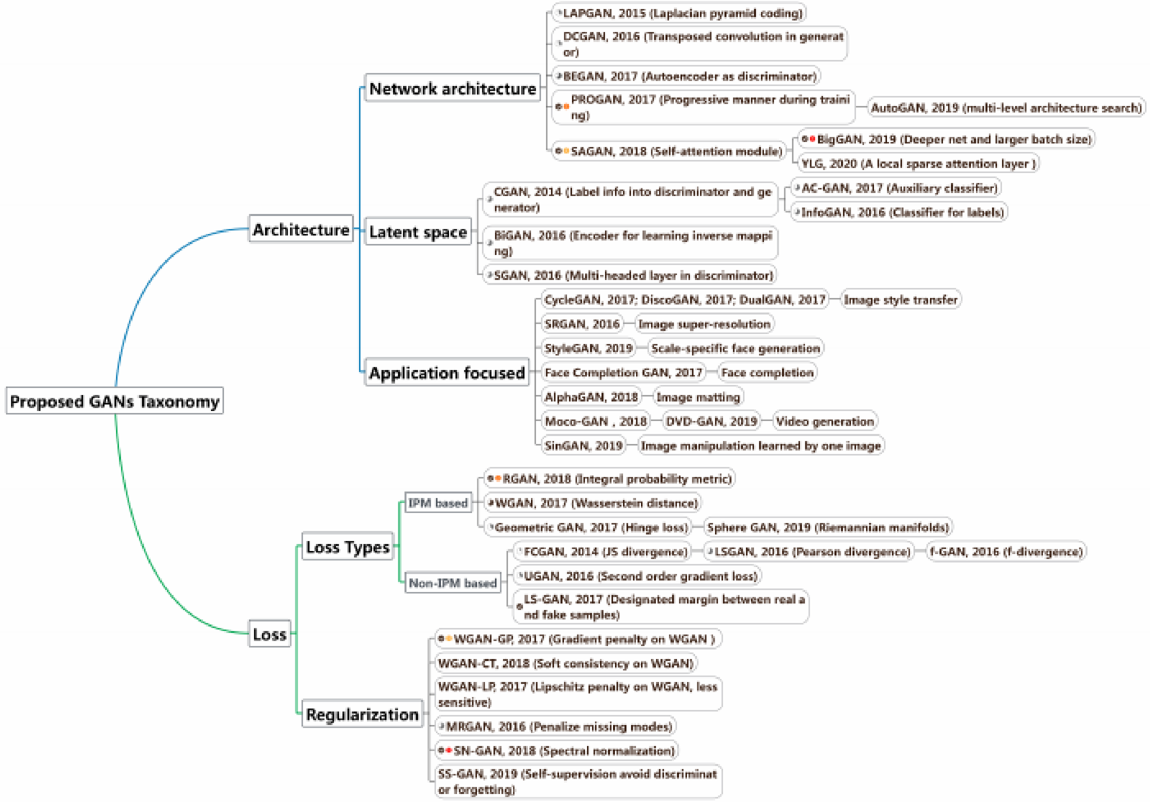
Find a latent vector z that satisfies

$$\left|downsample\left(G(z)\right) - I_{input}\right| < \varepsilon$$

PULSE: https://openaccess.thecvf.com/content_CVPR_2020/papers/Menon_PULSE_Self-Supervised_Photo_Upsampling_via_Latent_Space_Exploration_of_Generative_CVPR_2020_paper.pdf

Generative Adversarial Networks

- GANs are in Active Research


Proposed GANs Taxonomy
Architecture
Network architecture
LAPGAN, 2015 (Laplacian pyramid coding)
DCGAN, 2016 (Transposed convolution in generator)
BEGAN, 2017 (Autoencoder as discriminator)
PROGAN, 2017 (Progressive manner during training)
AutoGAN, 2019 (multi-level architecture search)
SAGAN, 2018 (Self-attention module)
BigGAN, 2019 (Deeper net and larger batch size)
YLG, 2020 (A local sparse attention layer)
Latent space
CGAN, 2014 (Label info into discriminator and generator)
AC-GAN, 2017 (Auxiliary classifier)
InfoGAN, 2016 (Classifier for labels)
BiGAN, 2016 (Encoder for learning inverse mapping)
SGAN, 2016 (Multi-headed layer in discriminator)
Application focused
CycleGAN, 2017; DiscoGAN, 2017; DualGAN, 2017
Image style transfer
SRGAN, 2016
Image super-resolution
StyleGAN, 2019
Scale-specific face generation
Face Completion GAN, 2017
Face completion
AlphaGAN, 2018
Image matting
Moco-GAN , 2018
DVD-GAN, 2019
Video generation
SinGAN, 2019
Image manipulation learned by one image
Loss
Loss Types
IPM based
RGAN, 2018 (Integral probability metric)
WGAN, 2017 (Wasserstein distance)
Geometric GAN, 2017 (Hinge loss)
Sphere GAN, 2019 (Riemannian manifolds)
Non-IPM based
FCGAN, 2014 (JS divergence)
LSGAN, 2016 (Pearson divergence)
f-GAN, 2016 (f-divergence)
UGAN, 2016 (Second order gradient loss)
LS-GAN, 2017 (Designated margin between real and fake samples)
Regularization
WGAN-GP, 2017 (Gradient penalty on WGAN)
WGAN-CT, 2018 (Soft consistency on WGAN)
WGAN-LP, 2017 (Lipschitz penalty on WGAN, less sensitive)
MRGAN, 2016 (Penalize missing modes)
SN-GAN, 2018 (Spectral normalization)
SS-GAN, 2019 (Self-supervision avoid discriminator forgetting)

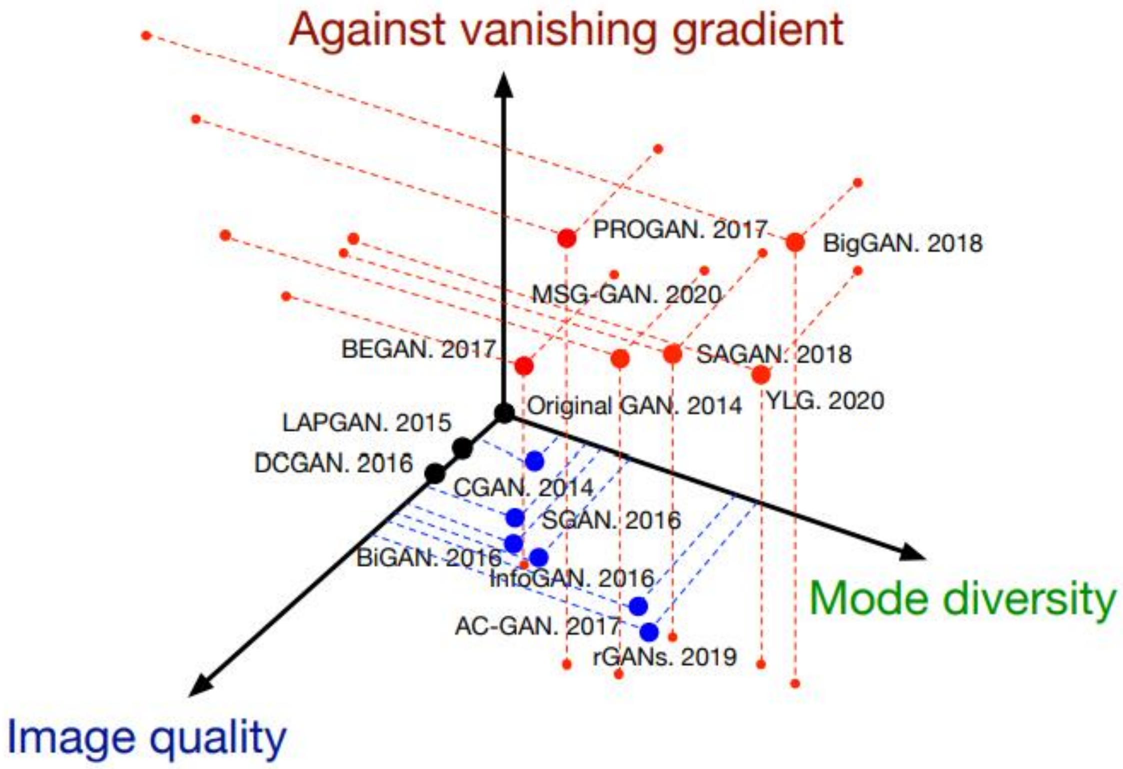


Image source: https://arxiv.org/pdf/1906.01529v6.pdf

商汤
sensetime

清華大学
Tsinghua University

Generative Image Models

Part 1 Introduction to Generative Image Models

Part 2 Explicit Density Models

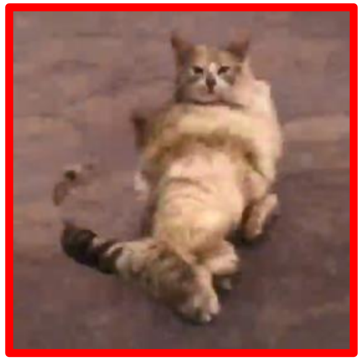
Part 3 Generative Adversarial Networks

Part 4 Generative Image Models Beyond GANs

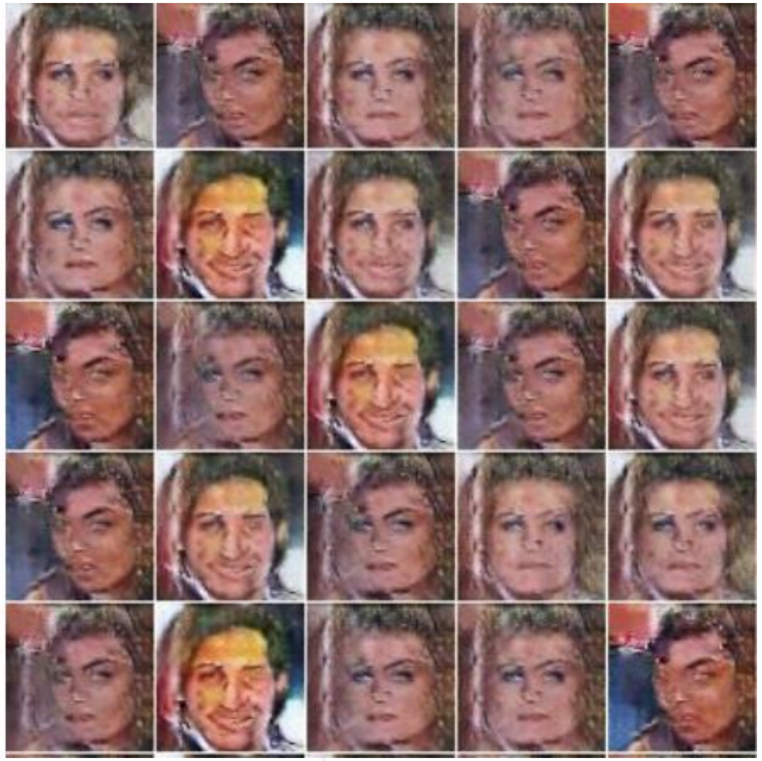
Generative Image Models Beyond GANs

GANs are Not Robust (Mode Collapse)

What kind of distributions do GAN learn?

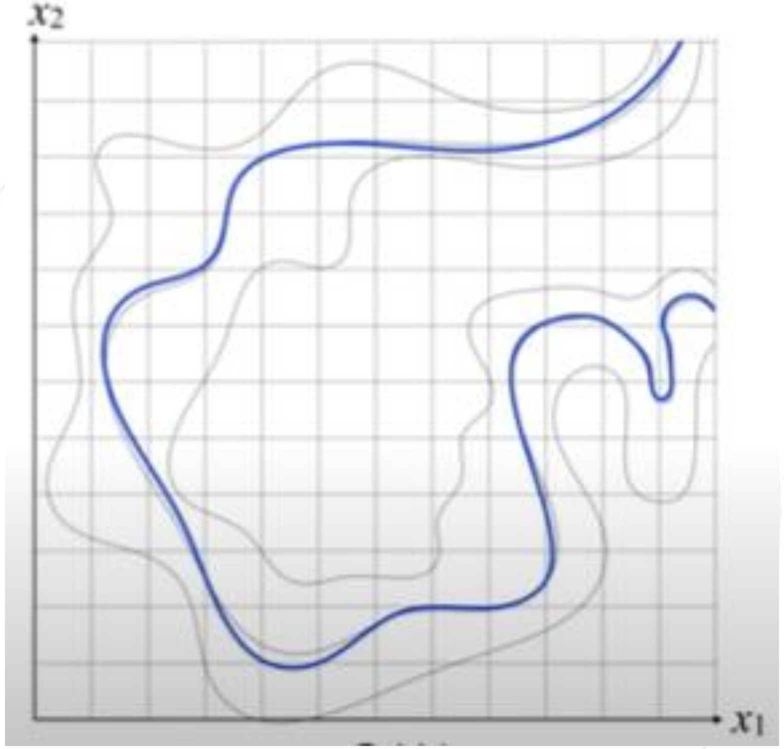


Failure samples



Limited modes

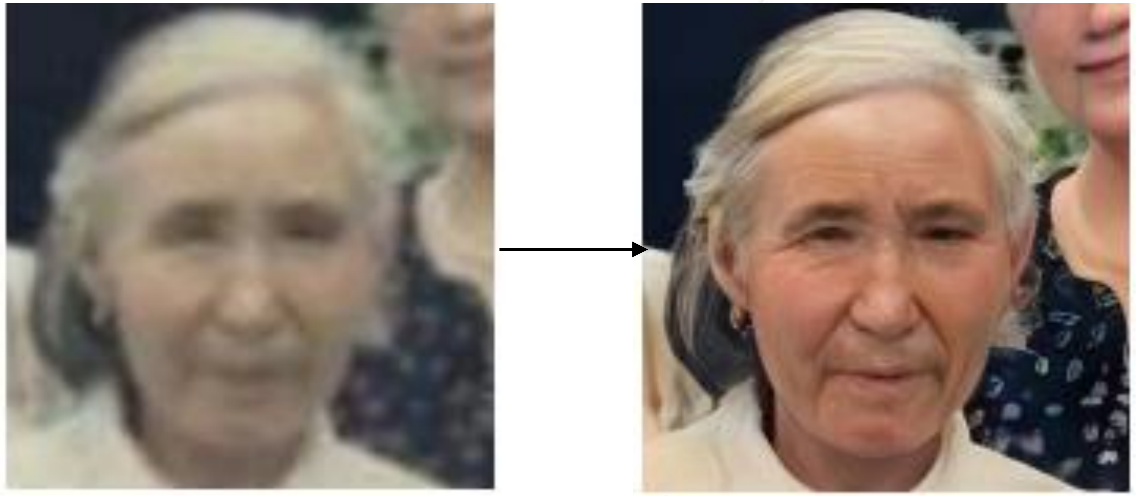
Image source: MIT 6.S191 (2018): Deep Generative Modeling



To satisfy GANs objective we only need to model a portion of P(x) (though with high visual quality) instead of the whole of it

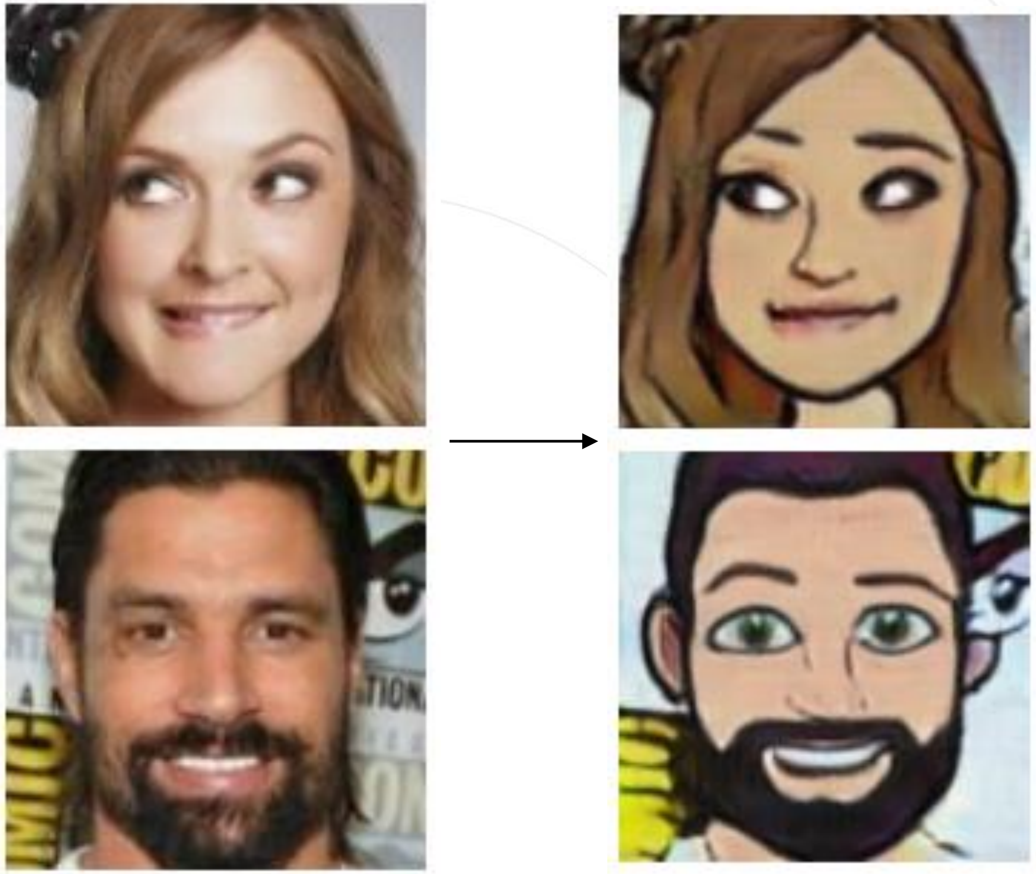
Generative Image Models Beyond GANs

- Industrial Applications of GANs are Limited (Mostly on Face Manipulation)



Face Restoration

Image source: https://arxiv.org/pdf/2101.04061.pdf



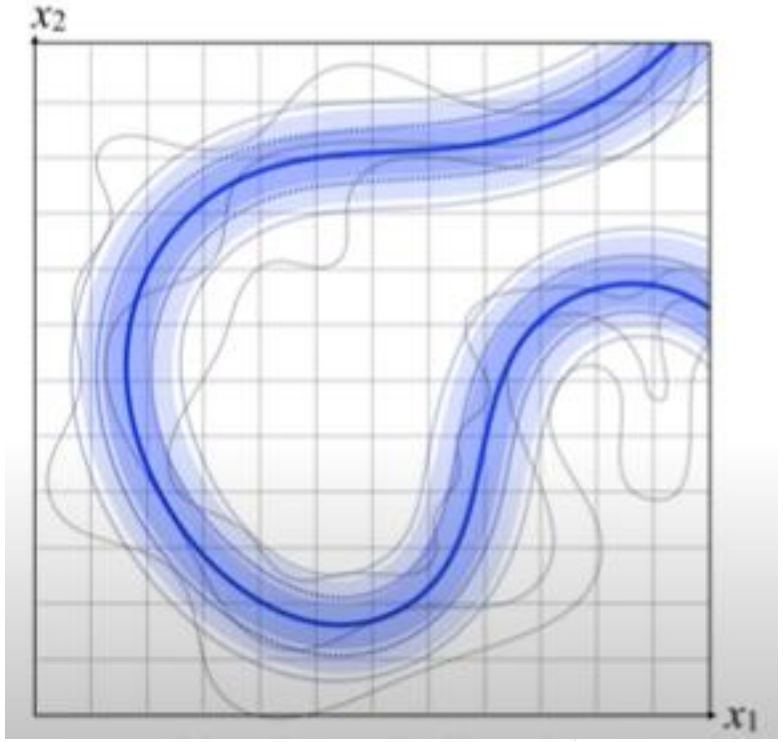
Real face to anime

Image source: https://arxiv.org/pdf/1907.01424.pdf

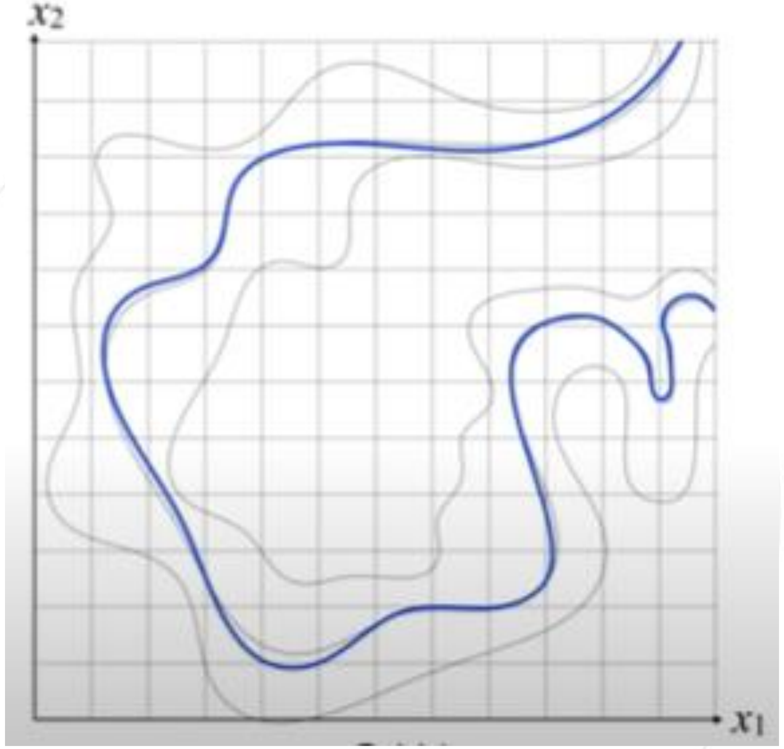
Generative Image Models Beyond GANs

■ GANs are Not Robust

Explicit Density Models are often more robust (they maximize the joint likelihood of the full dataset.)



To satisfy joint likelihood objective the probability mass needs to spread over the manifold



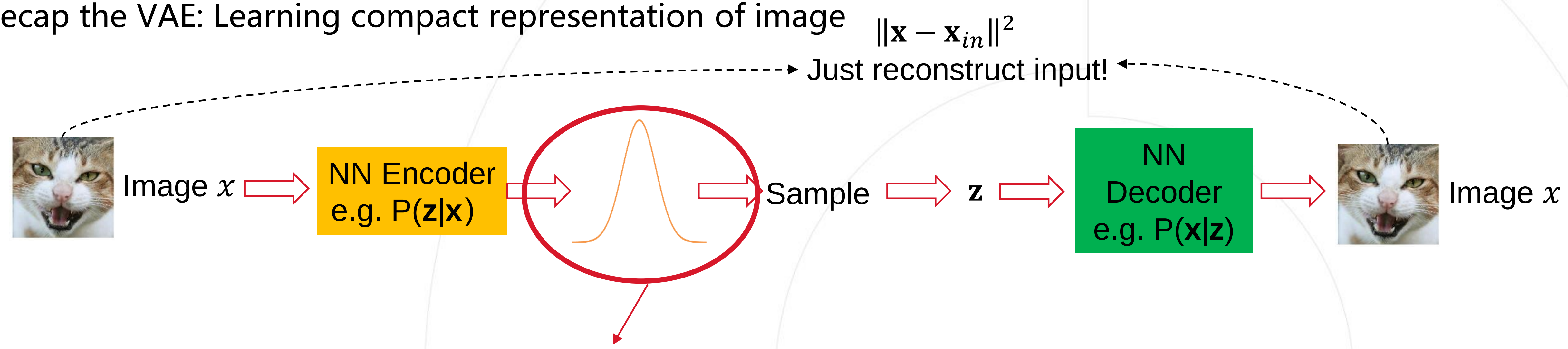
To satisfy GANs objective we only need to model a portion of P(x) (though with high visual quality) instead of the whole of it

Image source: MIT 6.S191 (2018): Deep Generative Modeling

Generative Image Models Beyond GANs

■ VQ-VAE

Recap the VAE: Learning compact representation of image

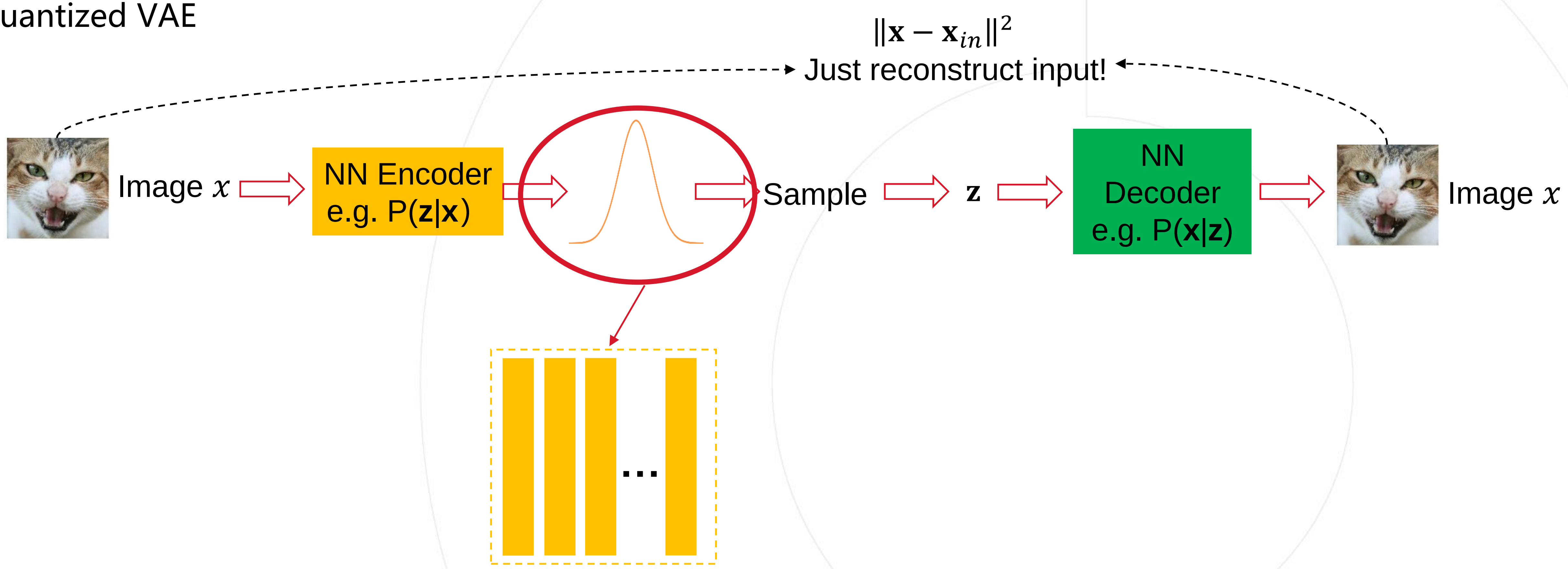


A continuous distribution (e.g. multi-variate Gaussian)

Generative Image Models Beyond GANs

VQ-VAE

Quantized VAE

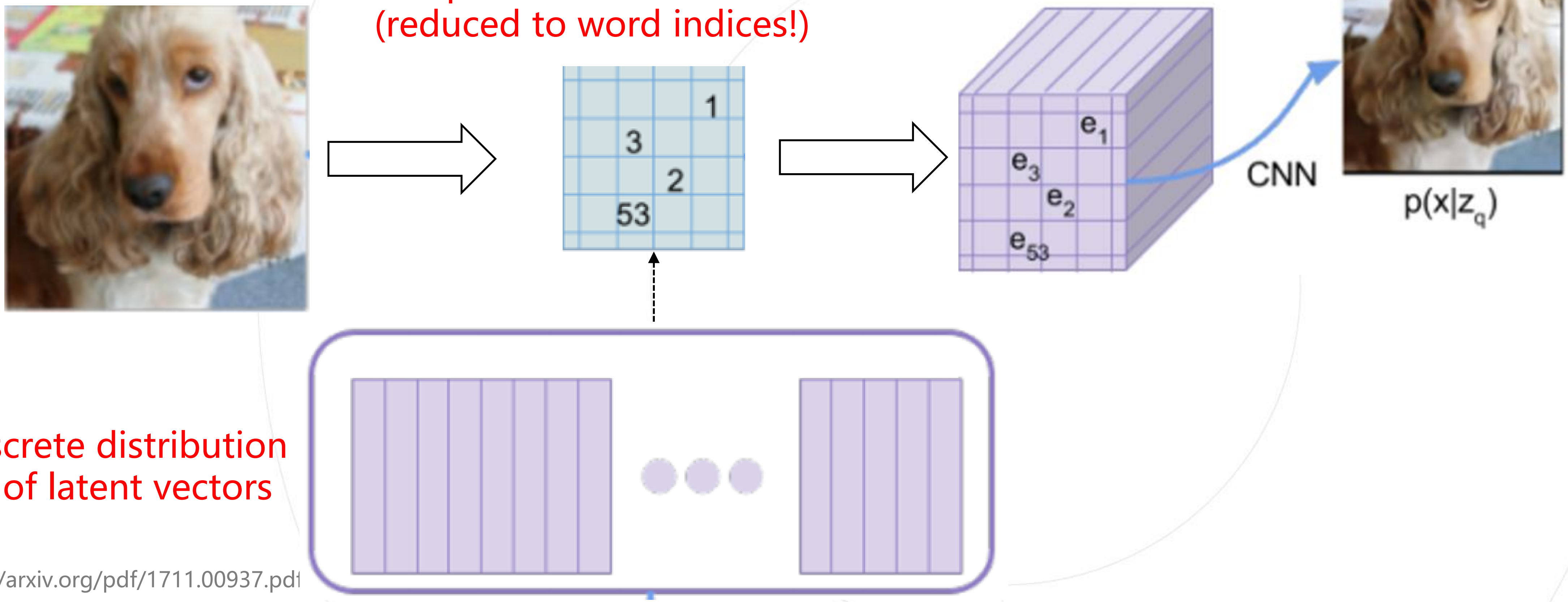


Replace the continuous distribution with a discrete one (i.e. a collection of latent vectors)

Generative Image Models Beyond GANs

■ VQ-VAE

Quantized VAE: Learning discrete latent representations



The latent representation is a spatial composition of words in dictionaries (reduced to word indices!)

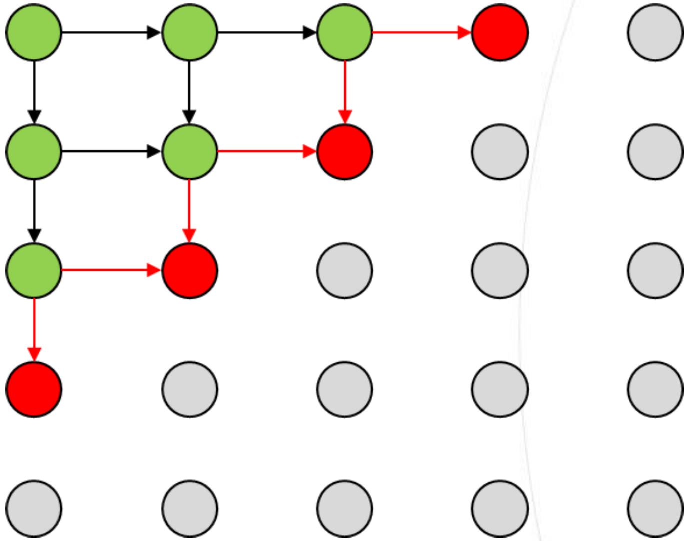
A discrete distribution (set) of latent vectors

Image source: https://arxiv.org/pdf/1711.00937.pdf

Generative Image Models Beyond GANs

- VQ-VAE

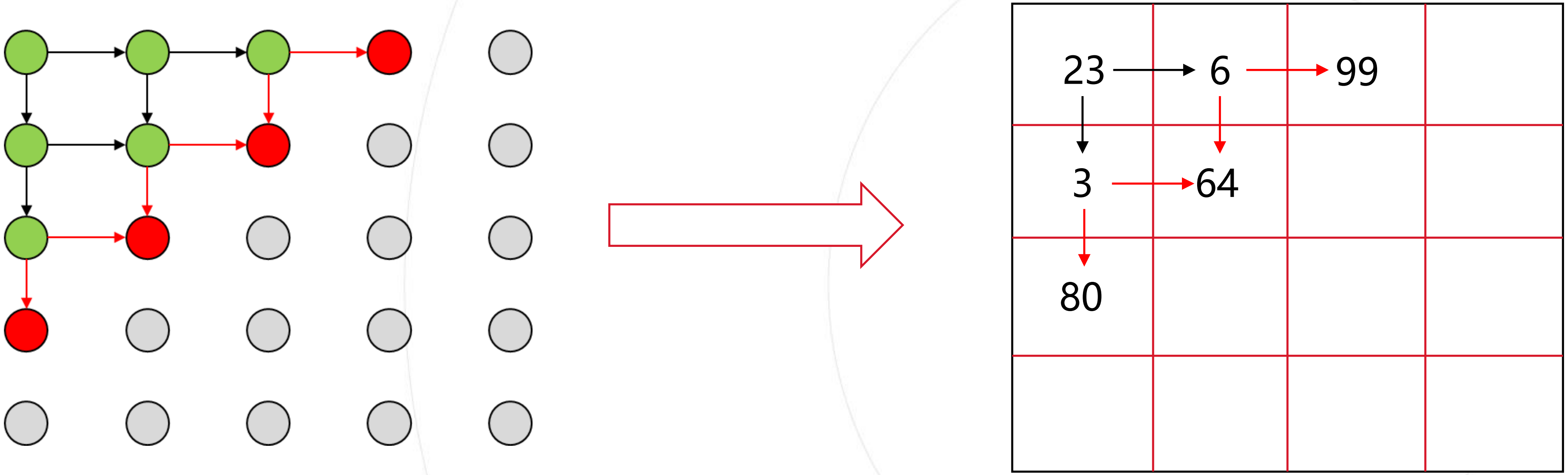
PixelRNN: Learns rich spatial relations among pixels, good sample quality, slow inference.



Generative Image Models Beyond GANs

■ VQ-VAE

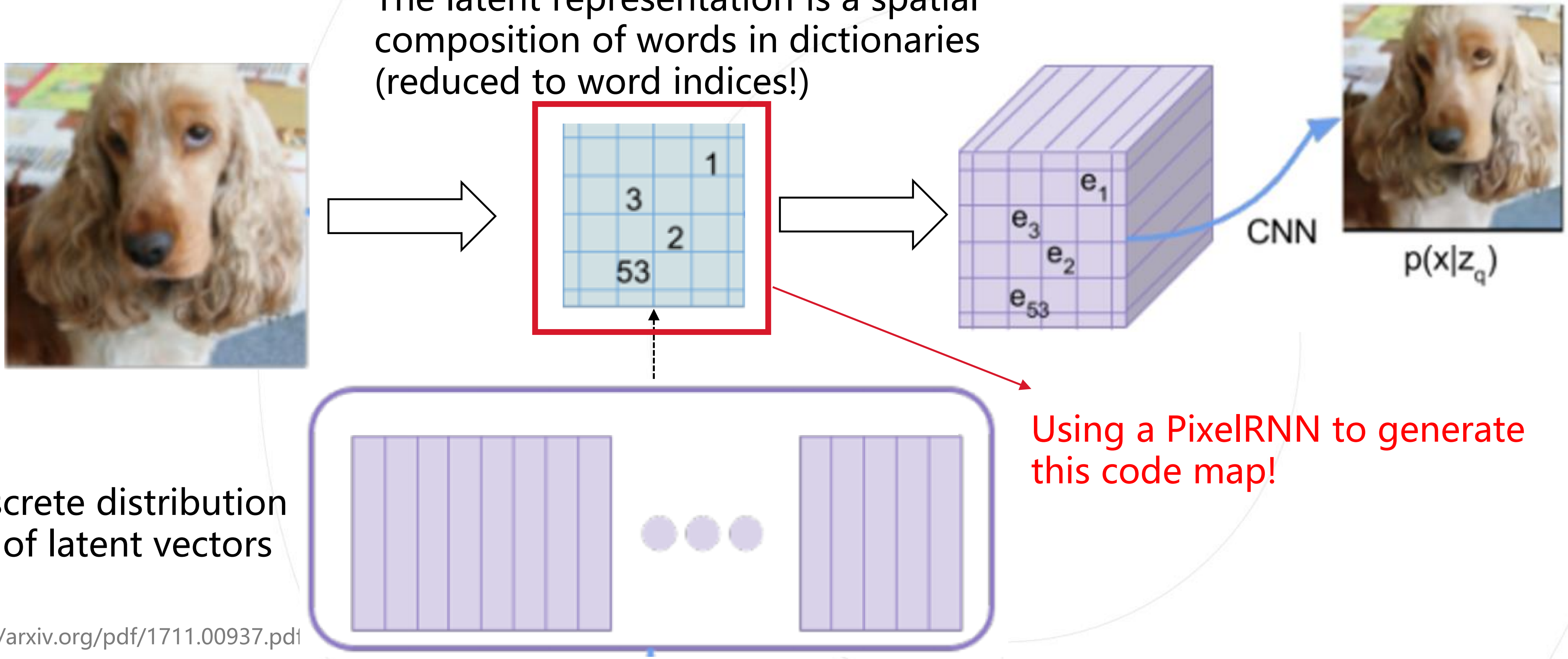
Learn spatial relations of ***image patches*** with PixelRNN. This makes sampling fast while inherits spatial modeling of PixelRNN



Generative Image Models Beyond GANs

■ VQ-VAE

Quantized VAE: Learning discrete latent representations



The latent representation is a spatial composition of words in dictionaries (reduced to word indices!)

A discrete distribution (set) of latent vectors

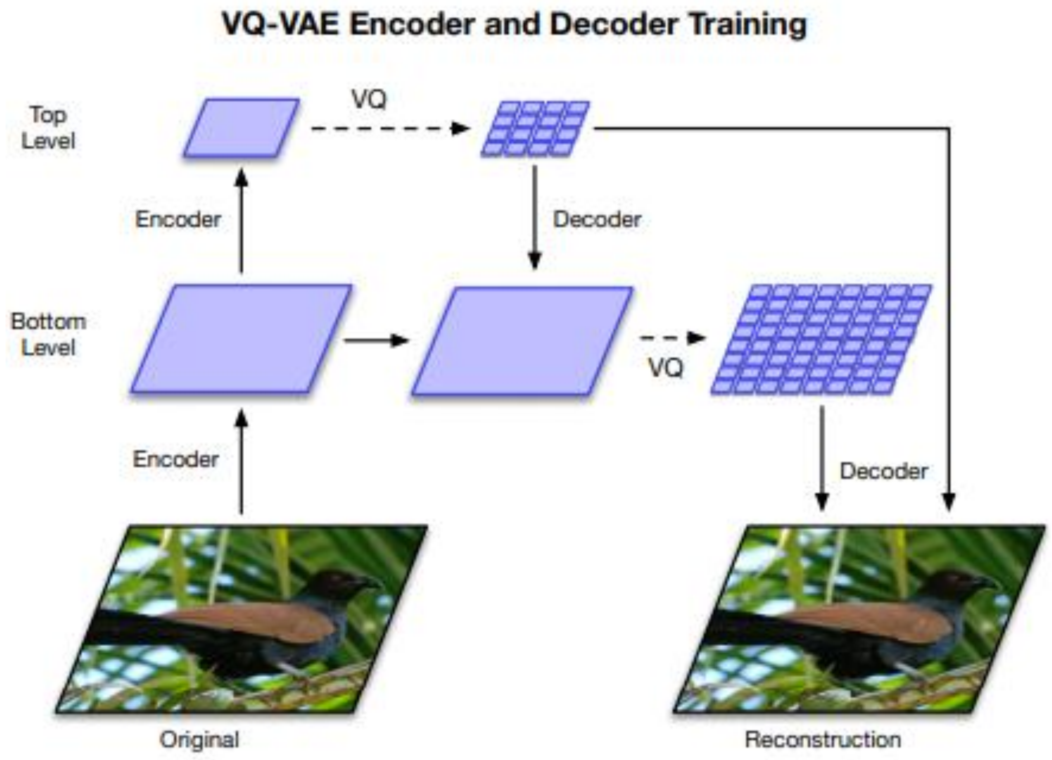
Using a PixelRNN to generate this code map!

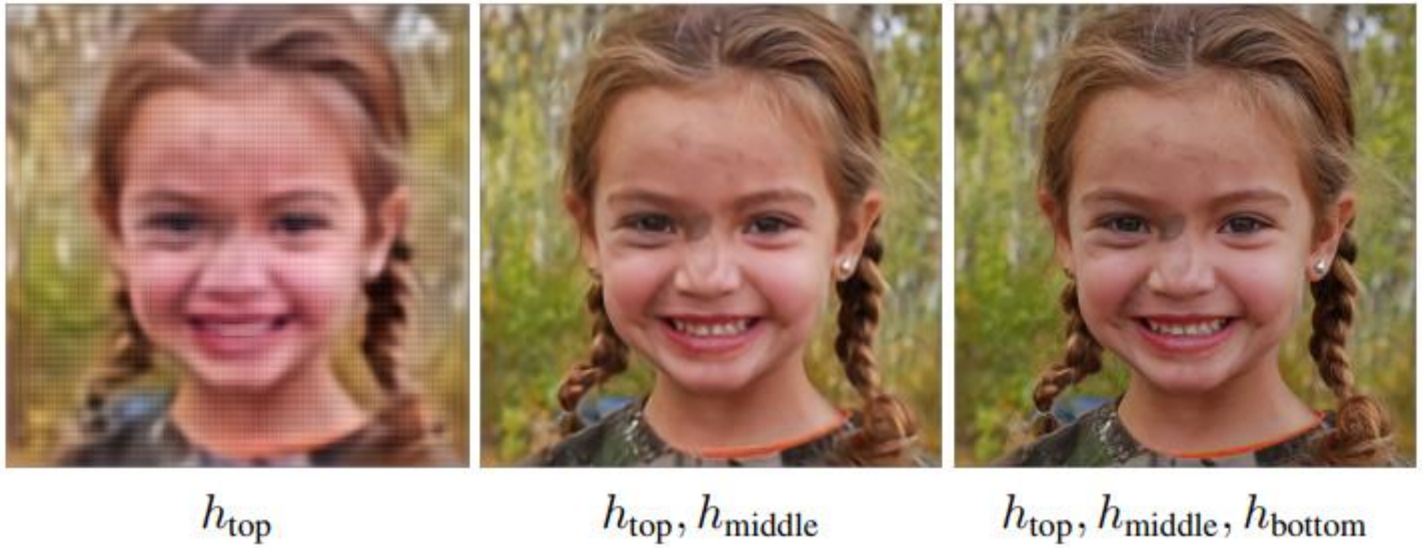
Image source: https://arxiv.org/pdf/1711.00937.pdf

Generative Image Models Beyond GANs

■ VQ-VAE

VQ-VAE-2: hierarchical VQ-VAE to generate high-resolution image





h_{top}
$h_{\text{top}}, h_{\text{middle}}$
$h_{\text{top}}, h_{\text{middle}}, h_{\text{bottom}}$

Image quality in par with GANs, but with no mode collapse!

Image copyright: https://openreview.net/pdf/ca805f55886316b9b980e0b981b4993675fc7e24.pdf

Generative Image Models Beyond GANs

Recap

- Generative image models learn the margin density P(x) of natural images (explicitly or implicitly)
- Explicit density models
  - PixelRNN Exact density evaluation, good samples, but inefficient training
  - Variational Autoencoders (VAE) Complex-to-simple prior transformation, blurry samples
  - Normalizing Flows Invertible prior transformation, limited modeling capacity
- Implicit Density Models
  - Generative Adversarial Networks Game-theoretic approach, best sampling quality. Not robust
- Generative Image Models Beyond GANs
  - VQ-VAE Combining modeling power of PixelRNN + efficient sampling of VAE. Very promising sample quality, no mode collapse

Generative Image Models Beyond GANs

Some useful resources

PixelRNN/PixelCNN:

- Pixel Recurrent Neural Network, van den Oord et al., ICML 2016
- Conditional Image Generation with PixelCNN Decoders, van den Oord et al., NIPS 2016
- PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications. Salimans et al., ICLR 2017

Varitional Autoencoders (VAE)

- Kinma and Welling, Auto-Encoding Variational Bayes, ICLR 2014
- Rezende, Mohamed and Wierstra, Stochastic Back-Propagation and Variational Inference in Deep Latent Gaussian Models, ICML 2014
- VAE tutorial: https://arxiv.org/abs/1606.05908

Normalizing Flows

- Tutorial from Li: https://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html
- Tutorial from Eric Zhang: https://blog.evjang.com/2018/01/nf1.html
- Glow from OpenAI: https://openai.com/blog/glow/

Generative Adversarial Networks (GAN)

- The official tutorial by Goodfellow: https://arxiv.org/abs/1406.2661
- A good step-by-step tutorial: https://github.com/zurutech/gans-from-theory-to-production#deep-diving-into-gans-from-theory-to-production
- A survey of GANs (advanced reading): https://arxiv.org/pdf/1906.01529v6.pdf

清華大學 Tsinghua University

商汤 sensetime

END