Image Deblurring and Dehazing

Na Li

vision@ouc

April 27.2017

Overview

1. Review of deblurring method
2. Introduction of dehazing method
3. Introduction of Convolution Neural Network
4. Dehazing by Convolution Neural Network

Fuzzy world

- Out of focus(Defocus blur)

Fuzzy world

- Out of focus(Defocus blur)

Fuzzy world

- Camera motion(Camera motion blur)

Fuzzy world

- Camera motion(Camera motion blur)

Fuzzy world

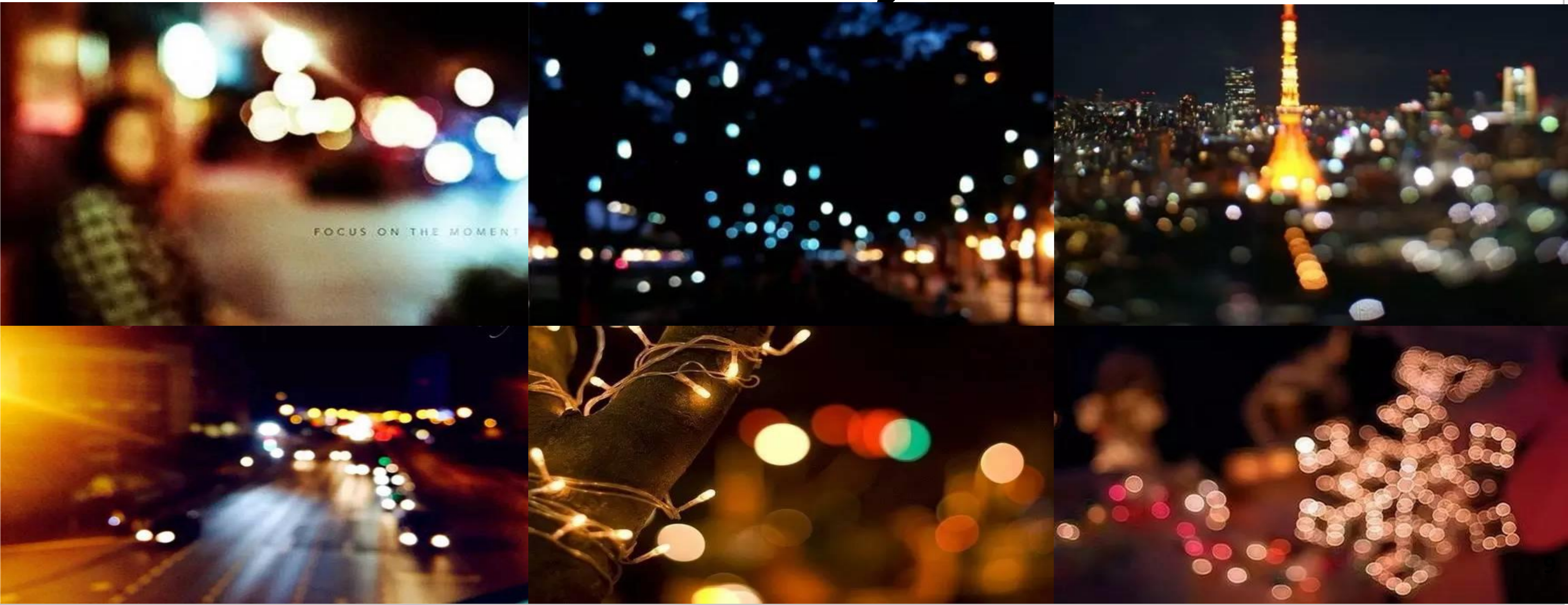
- Object movement(Object motion blur)

Fuzzy world

- Object movement(Object motion blur)

Blur

- **Often degrades image quality severely**
- **Unavoidable under dim light circumstances**



How to deblur

- 1.Image enhancement
- 2.Image restoration
- 3.Super resolution reconstruction

Image Enhancement

- Spatial domain method
- The operation of the pixels in the image

$$g(x,y)=T[f(x,y)]$$

f(x,y)------Original image

T------Space conversion function

g(x,y)------Processed image

Image Enhancement

➢ Frequency domain method(频域方法)

- Spatial domain---->

Frequency domain(transfer function)---->

Spatial domain

- Lowpass filtering------Smooth image------remove noise

- Highpass filtering------Sharpen image------remove blur and show the edge

- Homomorphic filtering------detail enhancement and contrast enhancement

Raw

Lowpass filtering

Highpass filtering

Homomorphic filter

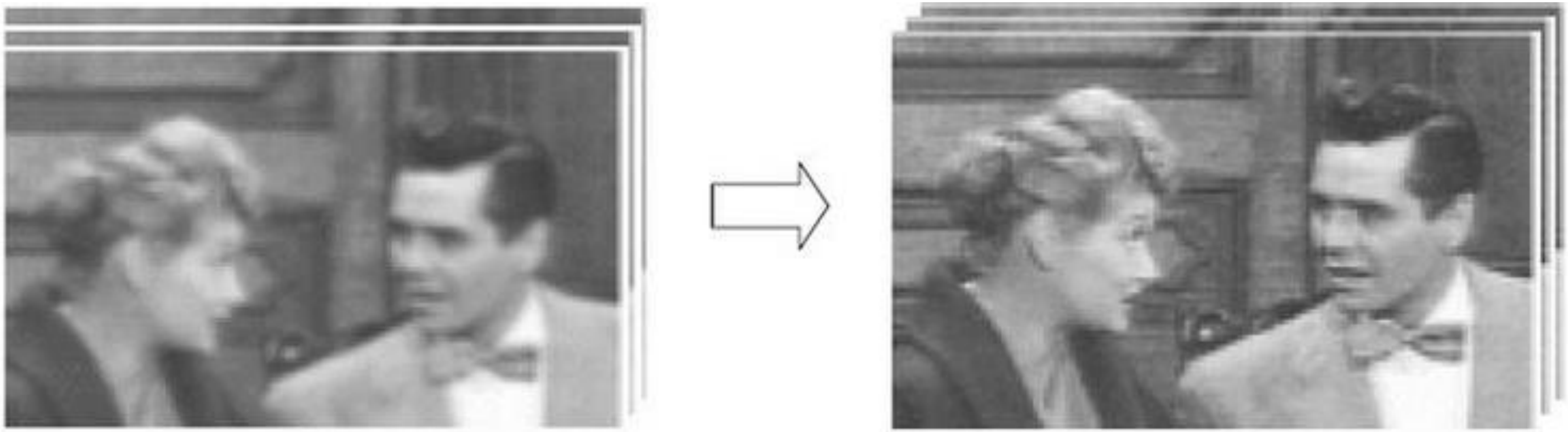
Image Restoration

- Adapt to: Defocus blur Camera motion blur Object motion blur

- $g(u,v)=h(u,v)*f(u,v)+n(u,v)$
- $G(u,v)=H(u,v)F(u,v)+N(u,v)$

Super-Resolution Reconstruction

- Low quality and low resolution---->High quality and high resolution



low resolution **low resolution**

Super-Resolution Reconstruction

➢Time resolution---->Spatial resolution

- Image super-resolution method based on reconstruction
- Image super-resolution method based on learning

Super-Resolution Reconstruction

- Principle:
  - Degradation model of image degradation

High resolution image

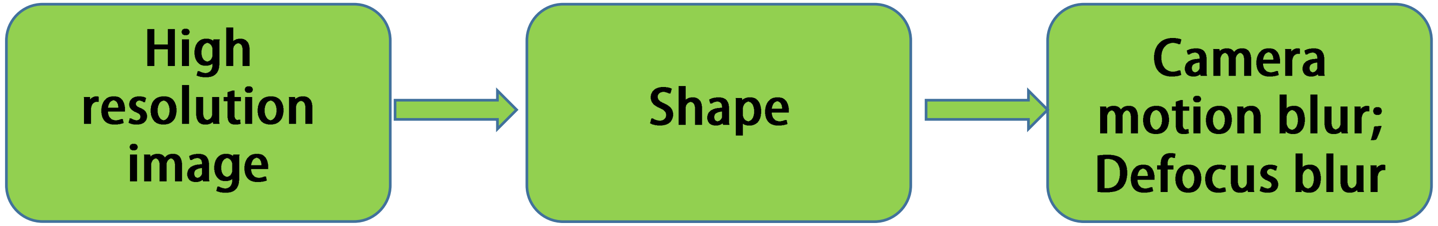
Super-Resolution Reconstruction

- Principle:
  - Degradation model of image degradation

High resolution image → Shape

Super-Resolution Reconstruction

- Principle:
  - Degradation model of image degradation



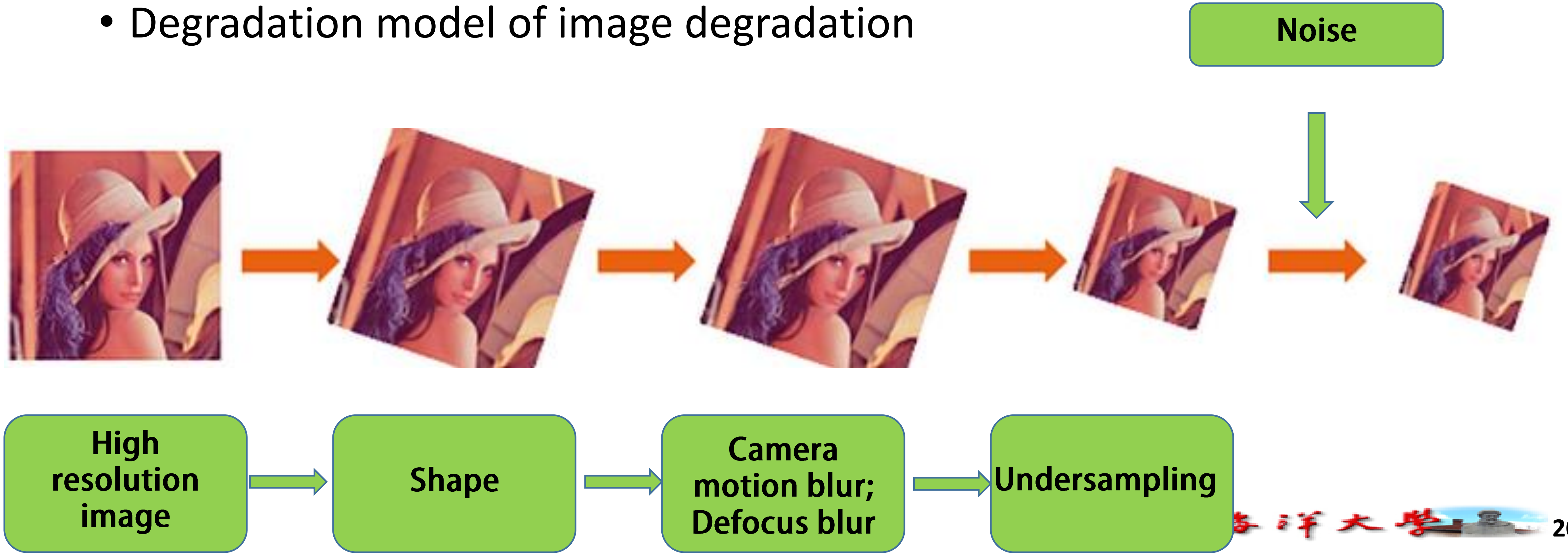
Super-Resolution Reconstruction

- Principle:
  - Degradation model of image degradation

High resolution image → Shape → Camera motion blur; Defocus blur → Undersampling

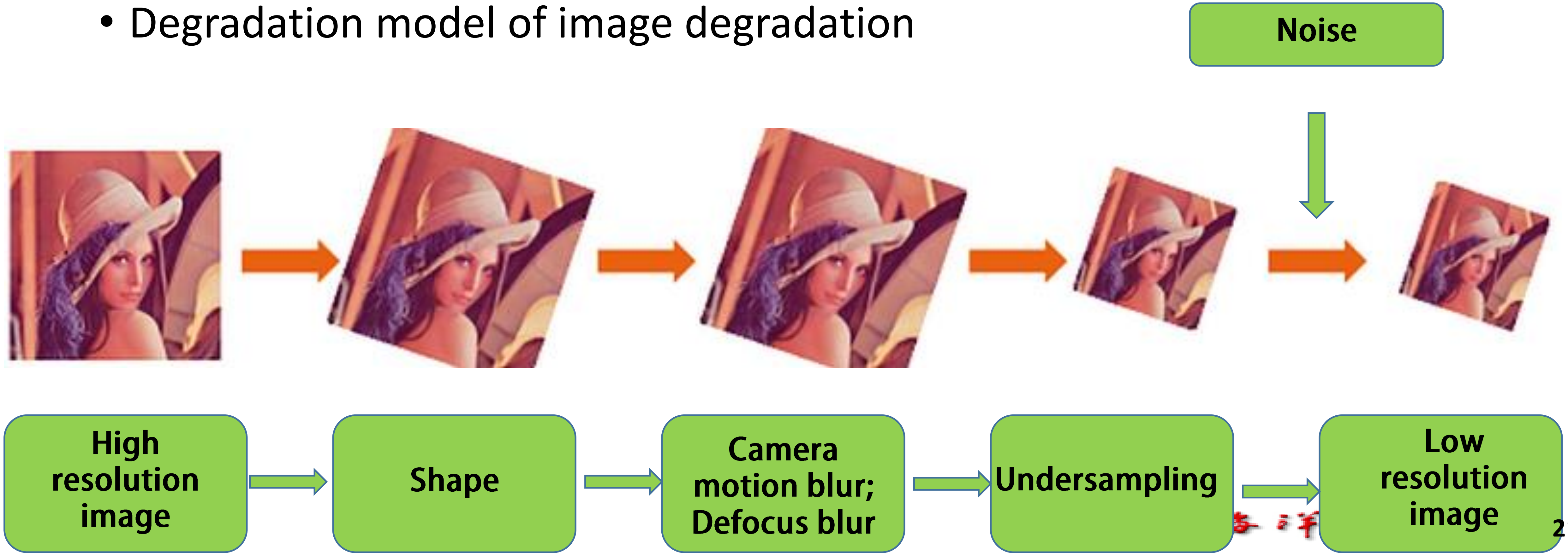
Super-Resolution Reconstruction

- Principle:
- Degradation model of image degradation



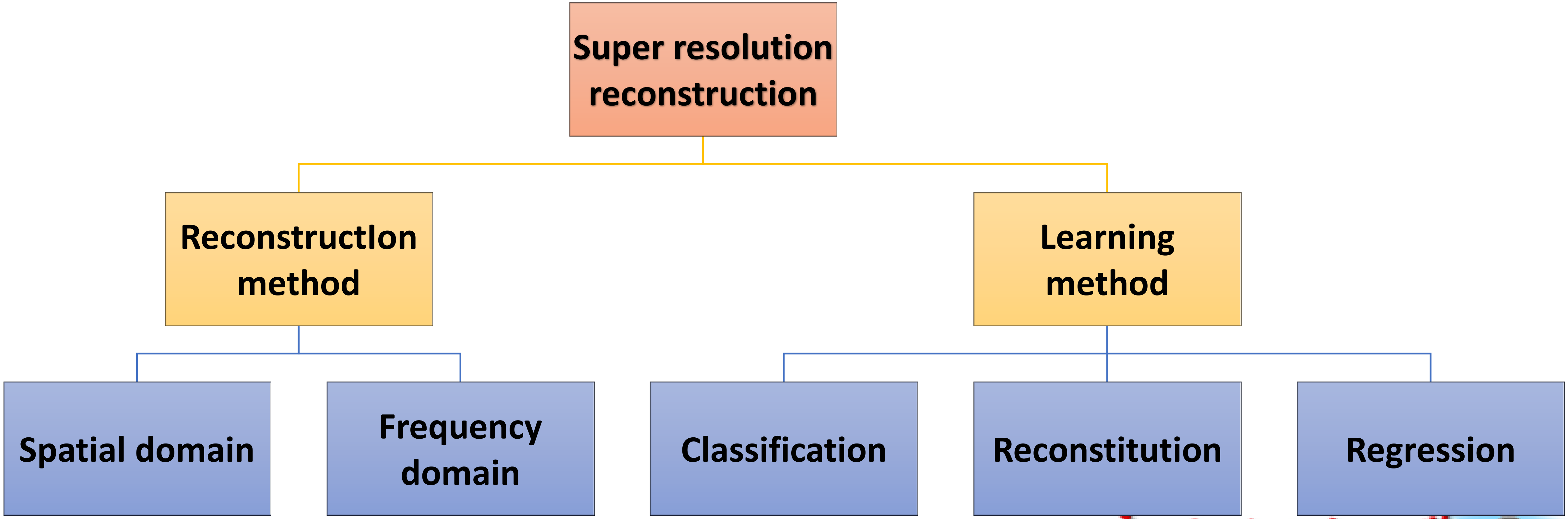
Super-Resolution Reconstruction

- **Principle:**
  - Degradation model of image degradation

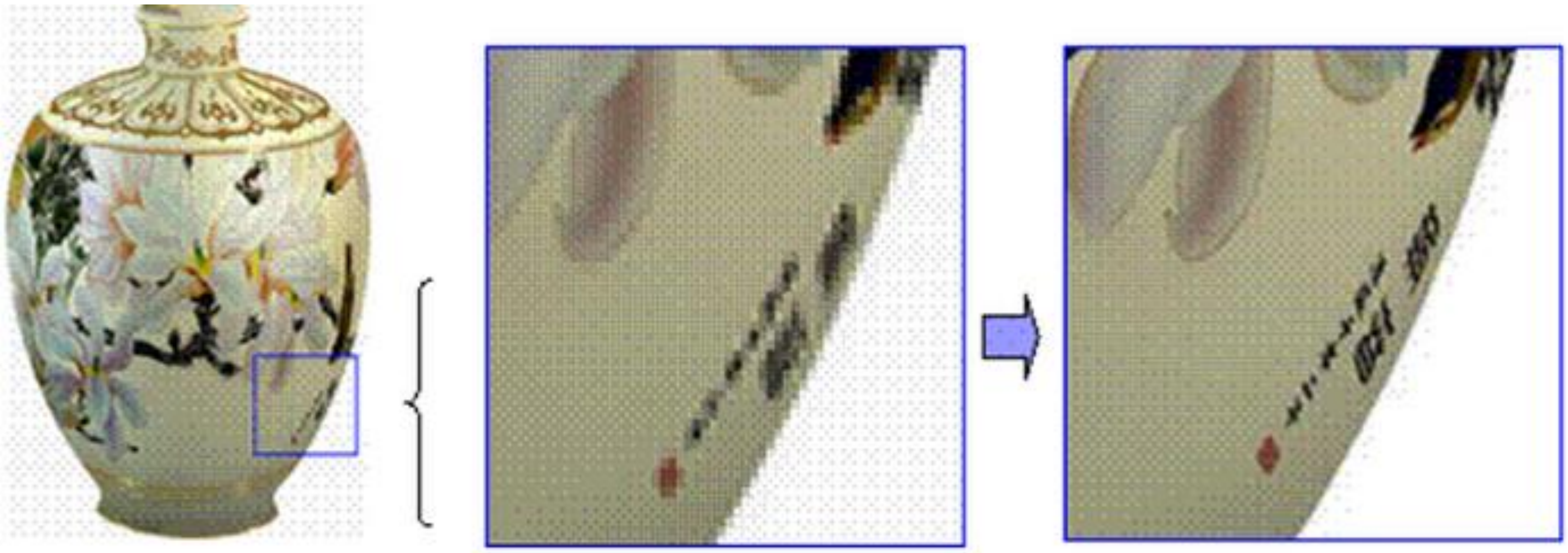


Super-Resolution Reconstruction

- Time resolution---->Spatial resolution



Super-Resolution Reconstruction

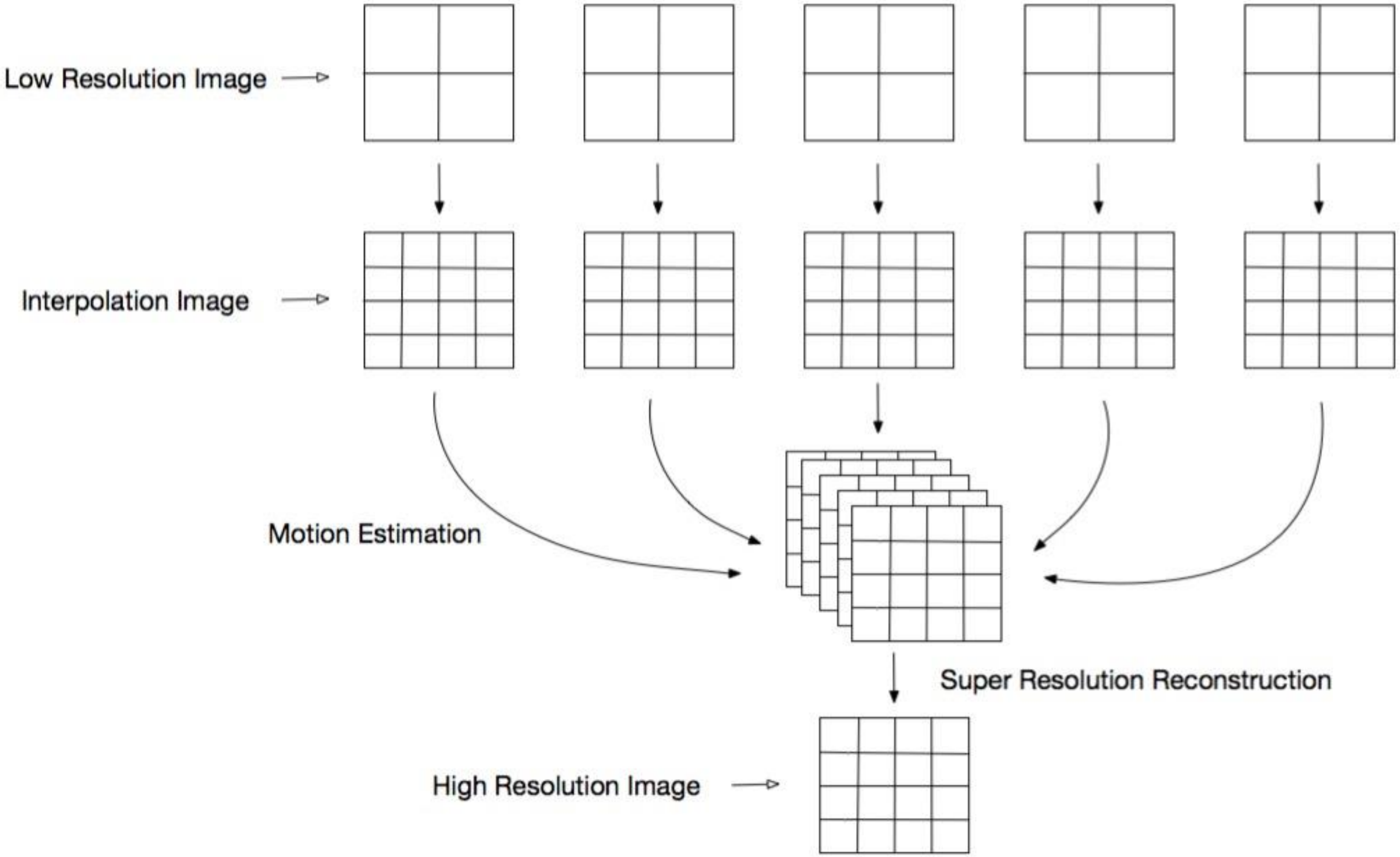


Super-Resolution Reconstruction

Super-Resolution Reconstruction

Super-Resolution Reconstruction

- Principle:
  - Super-resolution reconstruction
    - Inverse process



http://blog.csdn.net/geekmanong/article/details/46368509#comments

Super-Resolution Reconstruction

- **Advantage**

➢ Reconstruction method:

- Helpful to enhance and reconstruct of image spatial information

➢ Learning method:

- Aim to recovery of target categories
- Reduce computing
- Use pattern recognition and machine learning algorithms

Super-Resolution Reconstruction

- **Disadvantage**

➢ Reconstruction method:

- Difficult to deal with

➢ Learning method:

- Need lots of learning samples
- Need long time to training

Deblurring

- 1.Image enhancement
- 2.Image restoration
- 3.Super-resolution reconstruction

Changeable Weather






Haze/Fog Weather

Haze/Fog Weather

Haze/Fog Weather

Haze/Fog Weather

Haze

- **What is haze?**
  - **The characteristics of atmospheric degradation.**

Haze

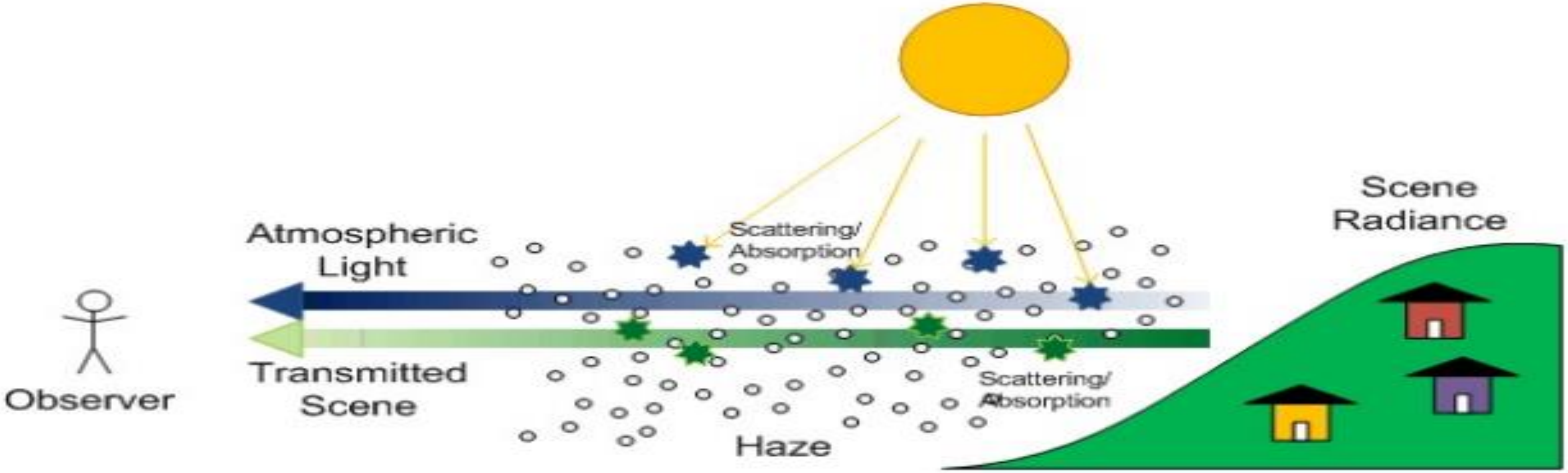
- **What is haze?**
  - **The characteristics of atmospheric degradation.**
  - **As distance between an object and the observer increases, atmosphere color replaces the color of the object.**

Haze

- **What is haze?**
  - **The characteristics of atmospheric degradation.**
  - **As distance between an object and the observer increases, atmosphere color replaces the color of the object.**
  - **Physical interactions of light with particles can be classified mainly as scattering and absorption.**

Haze

- **What is haze?**
- **Physical interactions of light with particles can be classified mainly as scattering and absorption.**



Application of Dehazing

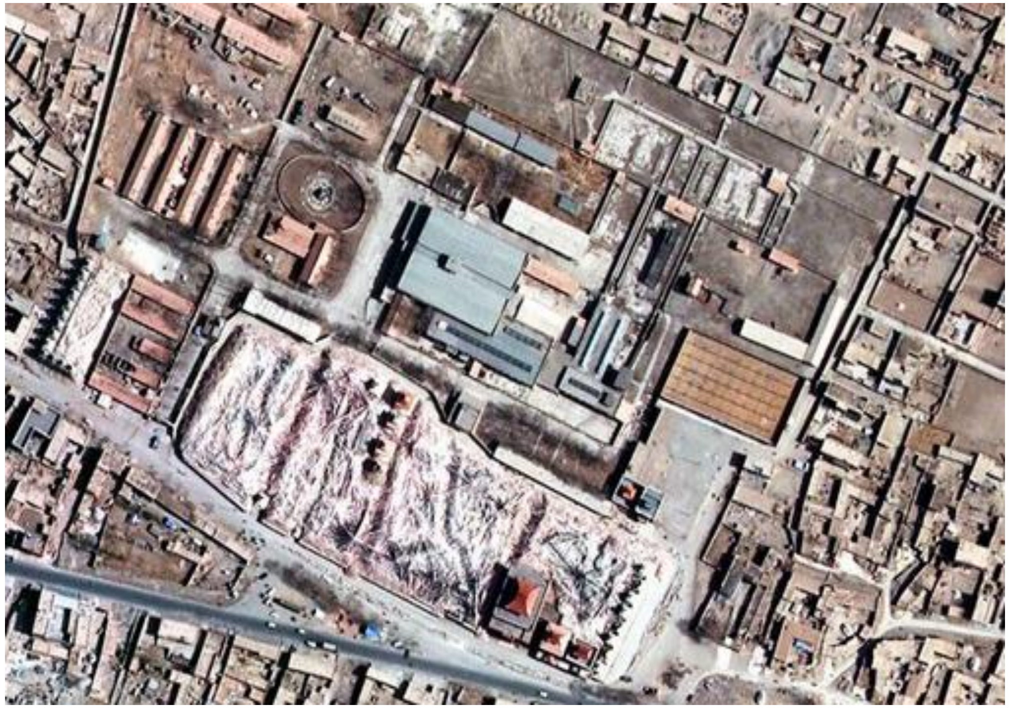
- **Removing this degradation is useful in many applications:**
- **Autonomous vehicle navigation**

Application of Dehazing

- **Removing this degradation is useful in many applications:**
- **Autonomous vehicle navigation**
- **Video surveillance systems**

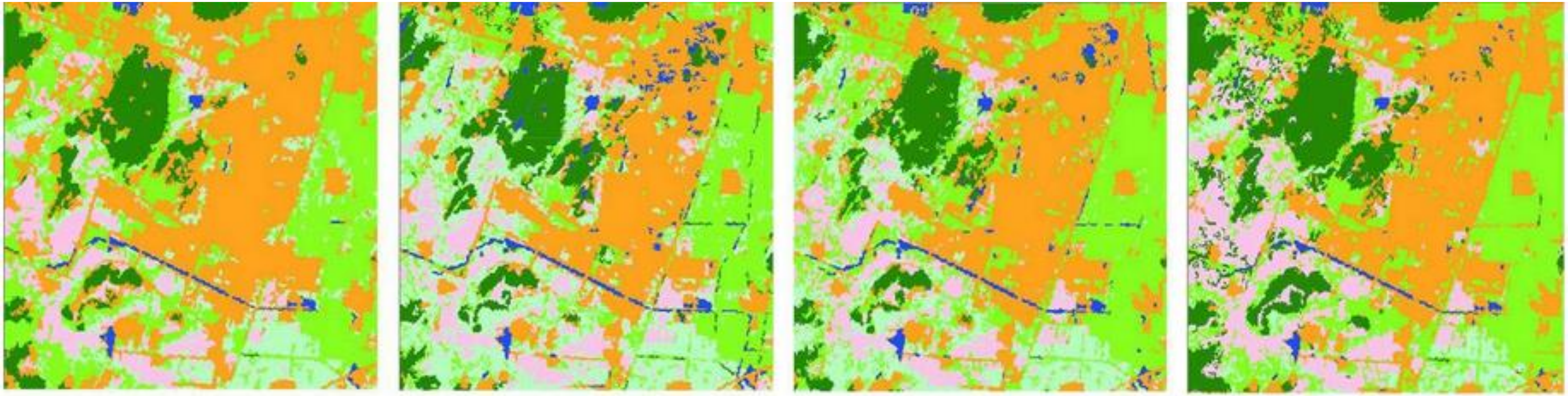
Application of Dehazing

- **Removing this degradation is useful in many applications:**
- **Autonomous vehicle navigation**
- **Video surveillance systems**
- **Aerial remote sensed images**



Application of Dehazing

- **Removing this degradation is useful in many applications:**
- **Land cover classification**



(a) 原多光谱图像分类图
(a) Classfiction figure of multi-spectral images

(b) Wavelet 融合分类图
(b) Classfiction figure of wavelet fusion

(c) Wavelet-HIS 融合分类图
(c) Classfiction figure of wavelet-HIS

(d) Wavelet-PCA 融合分类图
(d) Classfiction figure of wavelet-PCA fusion

农田 1 Farmland 1   农田 2 Farmland 2   水体 Water body   建设用地 Construction sites   裸地 Bare land   树 Tree

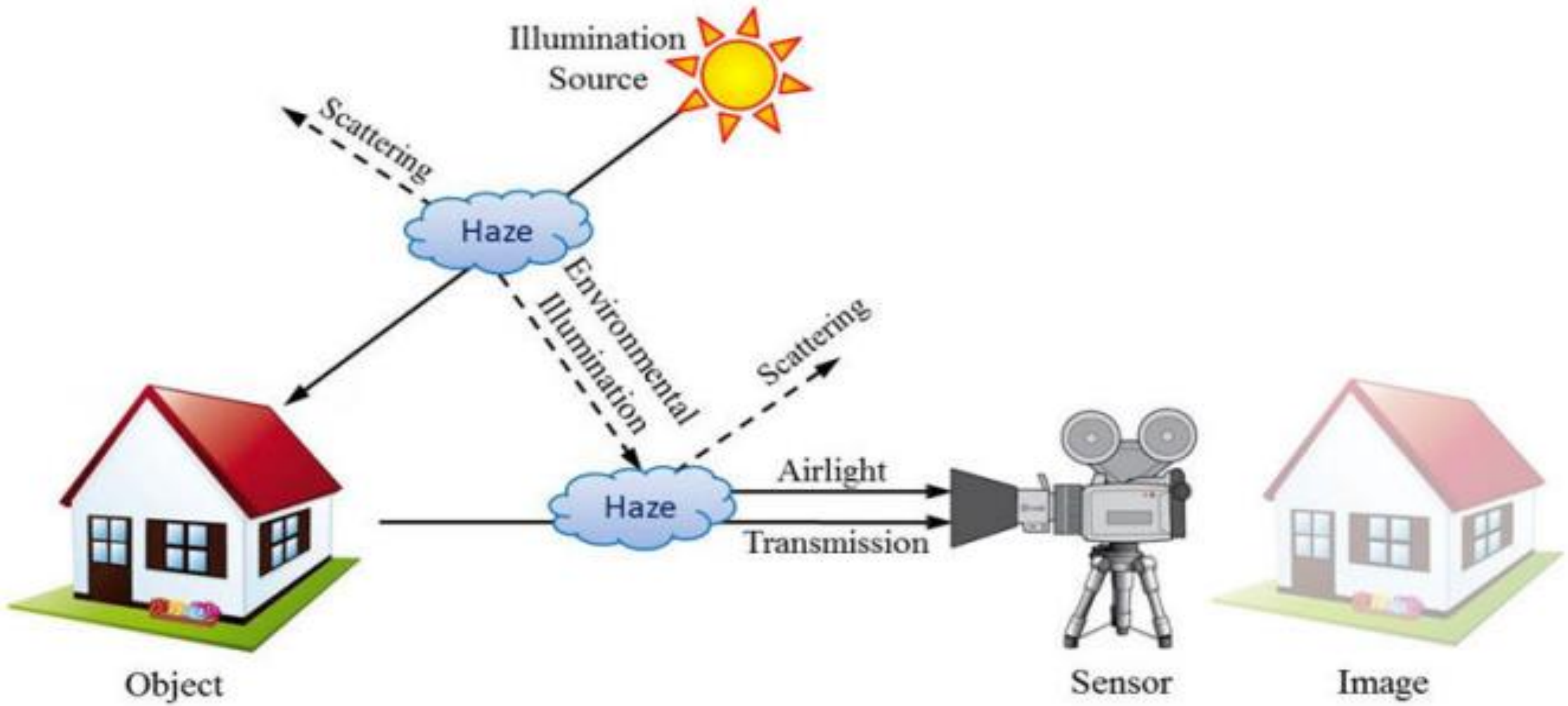
Application of Dehazing

- **Removing this degradation is useful in many applications:**
- **Land cover classification**
- **Underwater image restoration**



How to dehaze

- **Principle:**
- **The atmospheric scattering model**



How to dehaze

- **Principle:**
- **The atmospheric scattering model**

$$I(x) = J(x)t(x)+A(1-t(x))$$

  - **I(x) : the observed haze image**
  - **J(x) : the clear scene**
  - **t(x) : the medium transmission**
  - **A : the global atmospheric light(Airlight)**
  - **x : index pixels in the observed haze image**

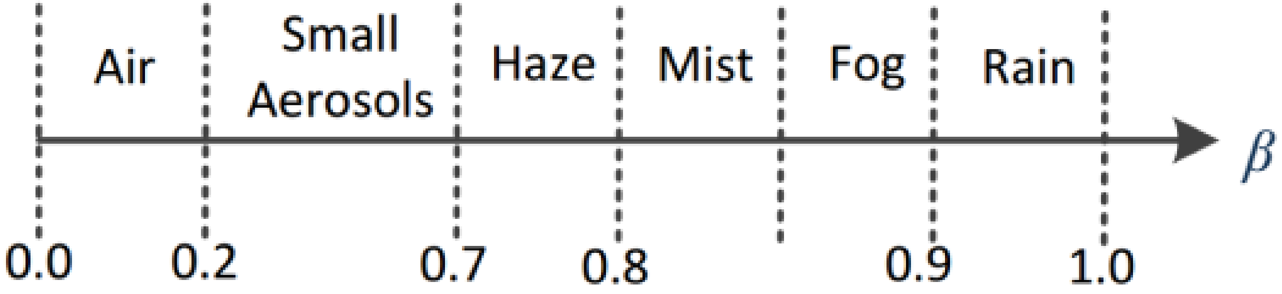
How to dehaze

- **Principle:**
- **The atmospheric scattering model**

$$\mathbf{I(x) = J(x)t(x)+A(1-t(x))}$$

$$\mathbf{t(x) = e^{-\beta d(x)}}$$

  - **d(x) : the distance from the scene point to the camera (depth map)**
  - **β : the scattering cofficient of the atmosphere**

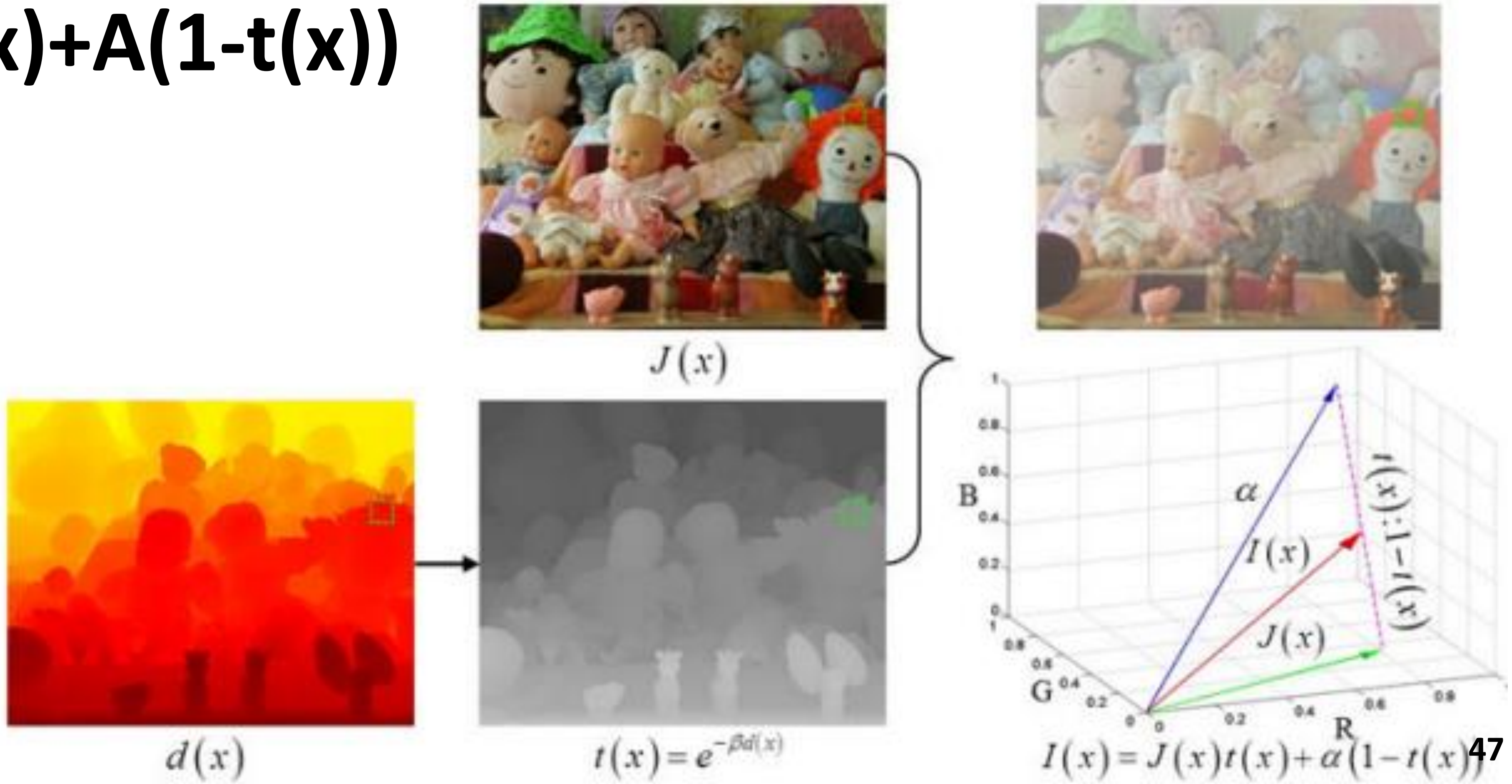


Srinivasa G. et al. Shedding Light on the Weather. CVPR 2003

How to dehaze

$$I(x) = J(x)t(x)+A(1-t(x))$$

$$t(x) = e^{-\beta d(x)}$$



How to dehaze

- **Principle:**
  - **The atmospheric scattering model**

$$I(x) = J(x)t(x)+A(1-t(x))$$

$$t(x) = e^{-\beta d(x)}$$

  - **Known conditions:** $I(x)$ (haze image)
  - **Unknown conditions:** $t(x)$ A
  - **Goal:** $J(x)$ (dehaze image)

How to dehaze

$$I(x) = J(x)t(x) + A(1 - t(x))$$

$$\Rightarrow J(x) = \frac{I(x) - A(1 - t(x))}{t(x)}$$

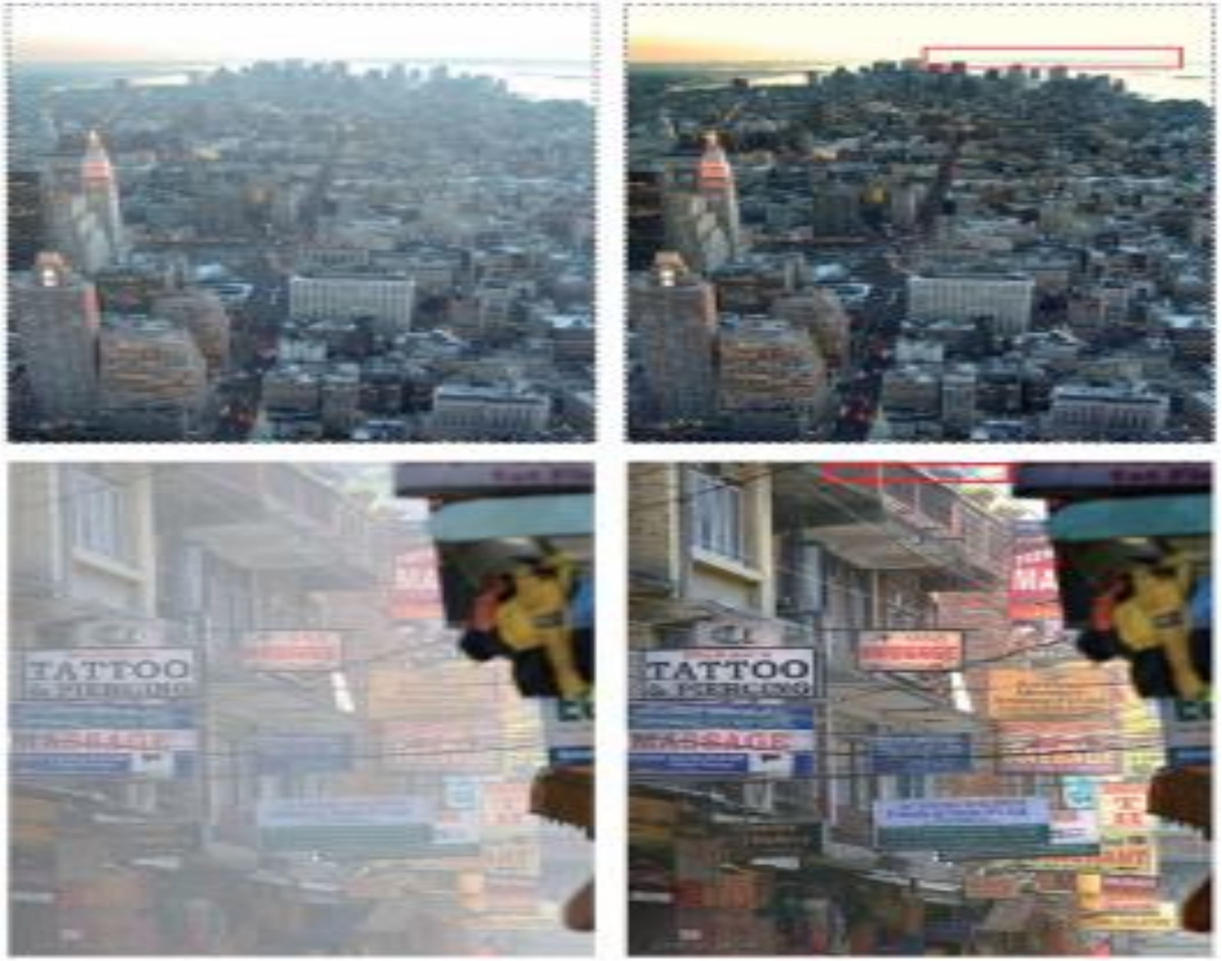
$$\Rightarrow J(x) = \frac{I(x) - A + At(x)}{t(x)}$$

$$\Rightarrow J(x) = \frac{I(x) - A}{t(x)} + A$$

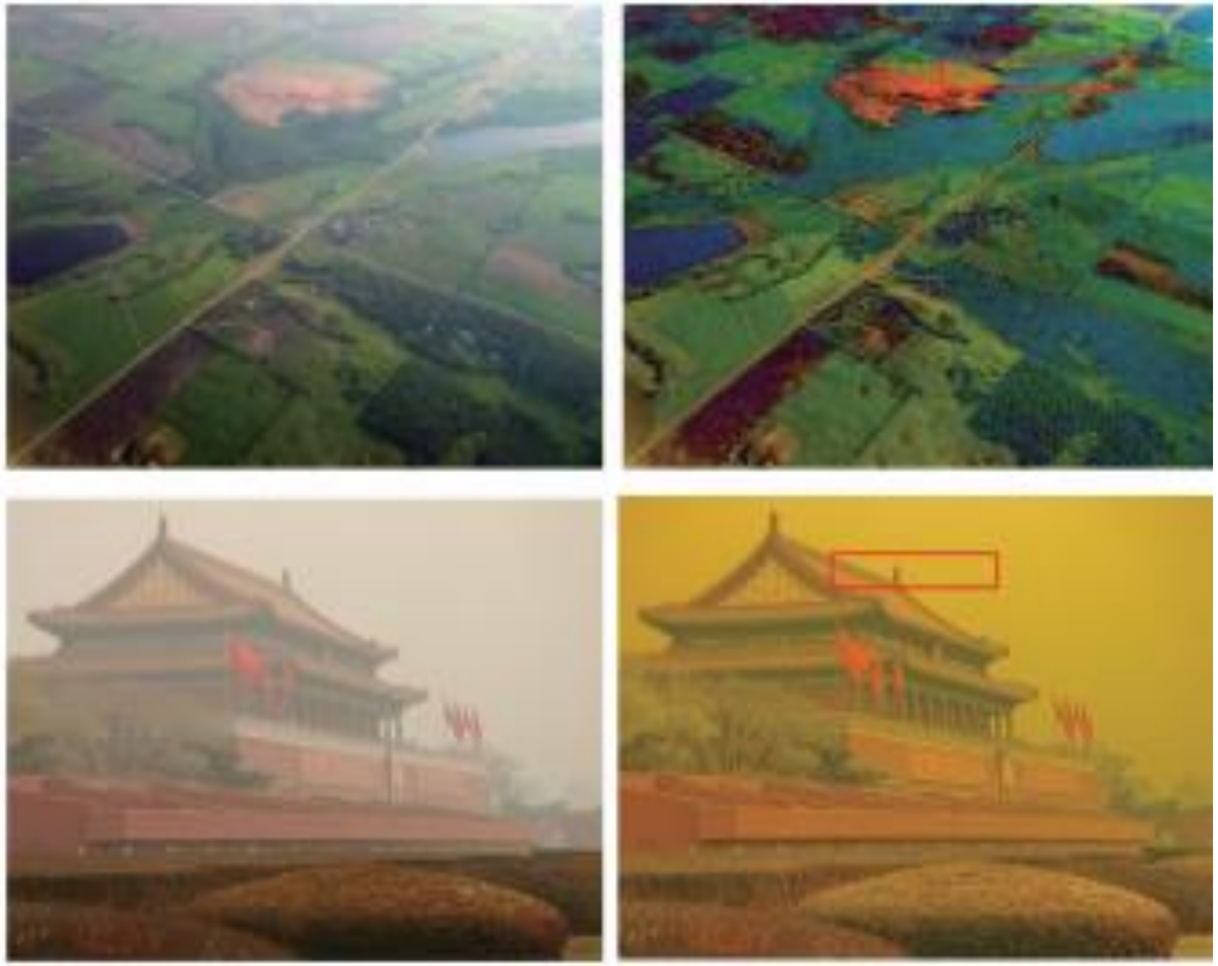
Image before processing---->Image after processing

Haze dataset

Haze dataset



Haze dataset

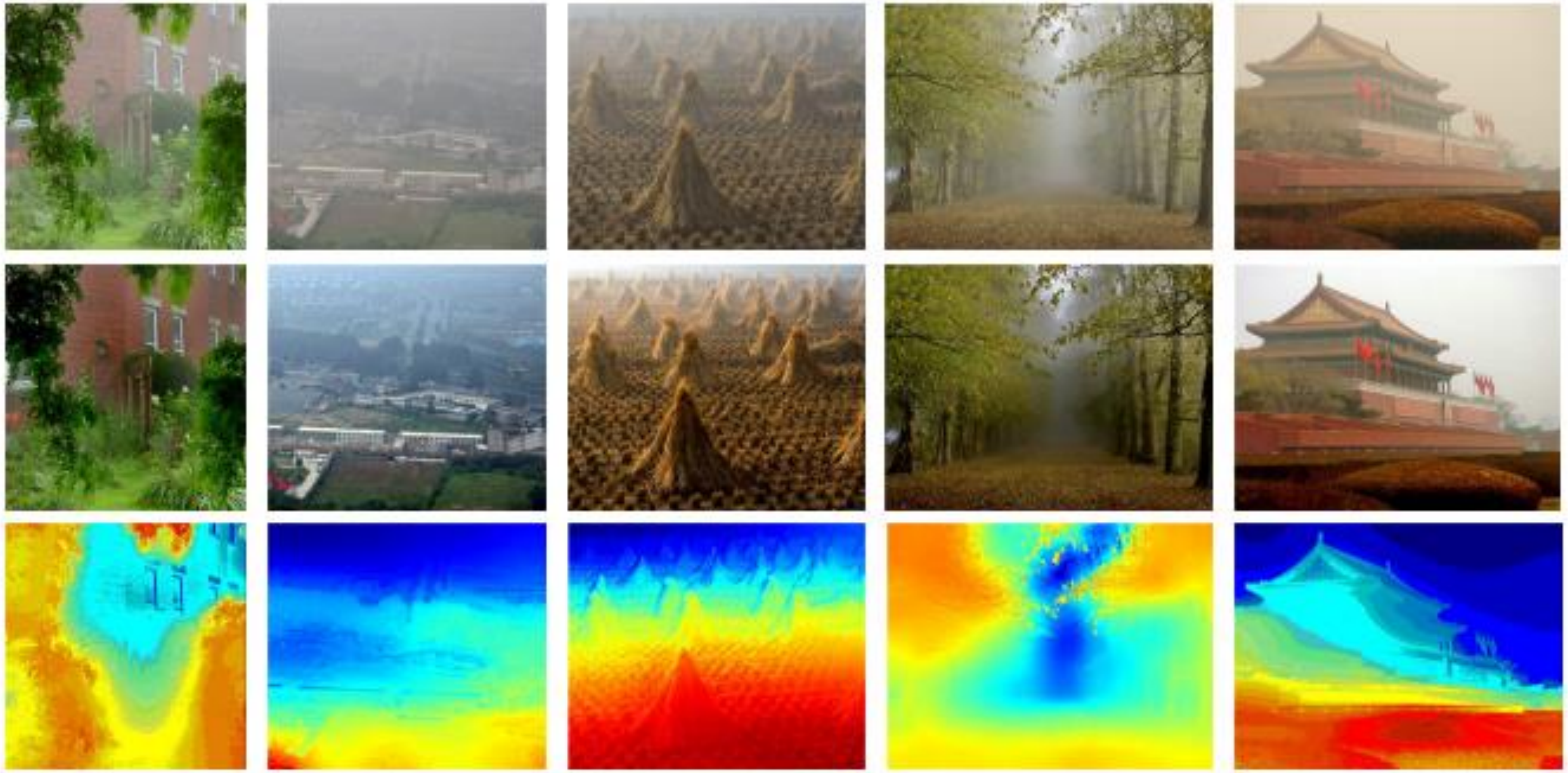


Haze dataset

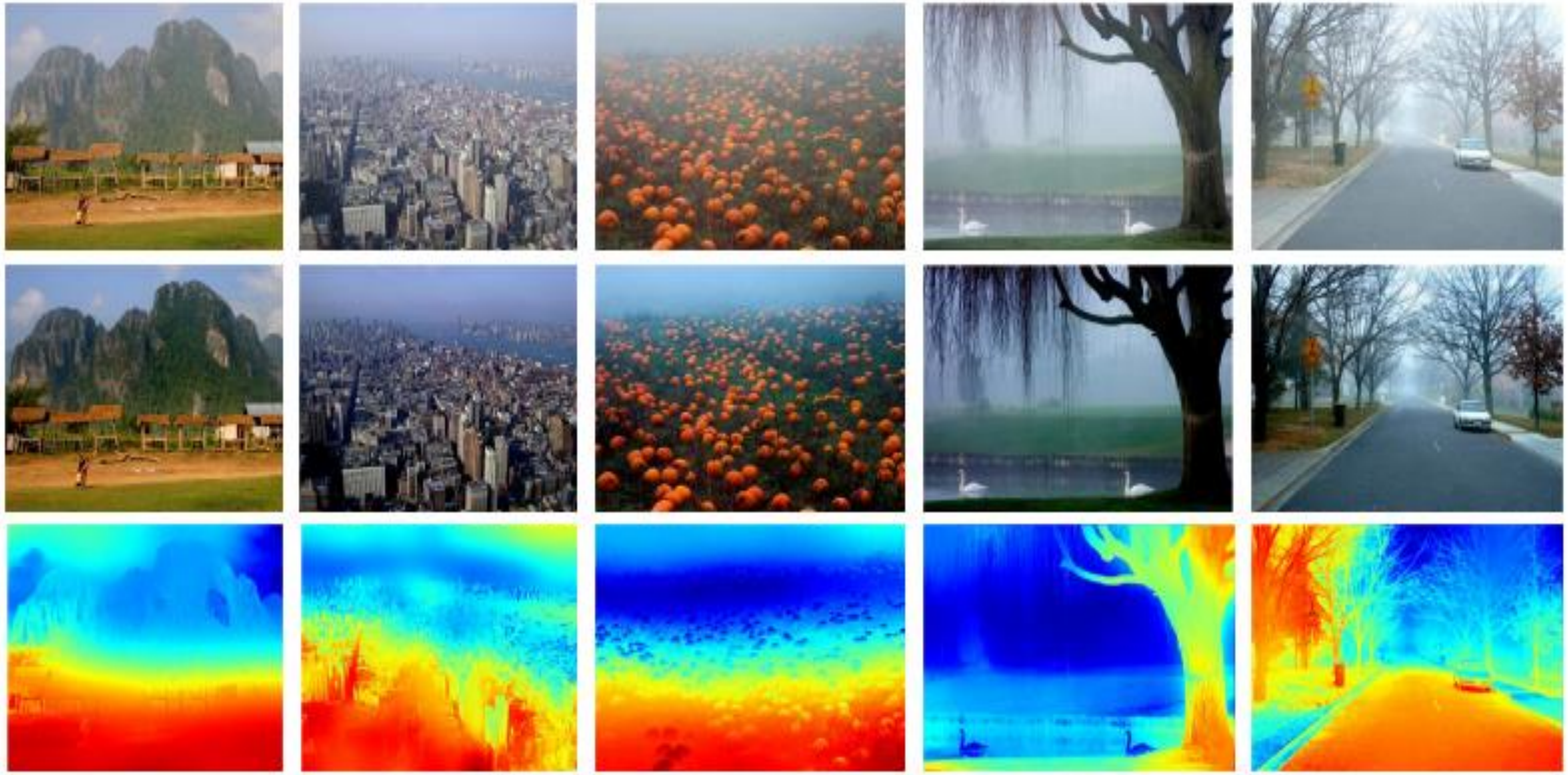
Haze dataset

Haze dataset

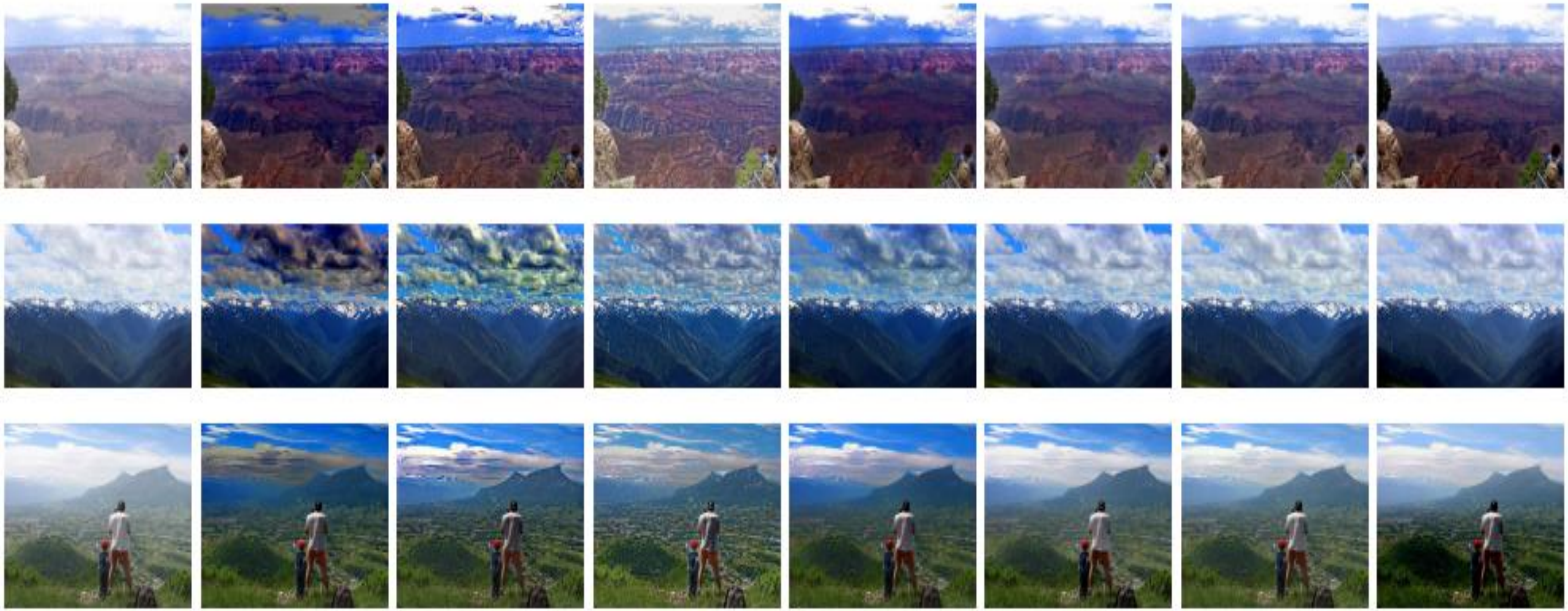
Haze dataset



Haze dataset

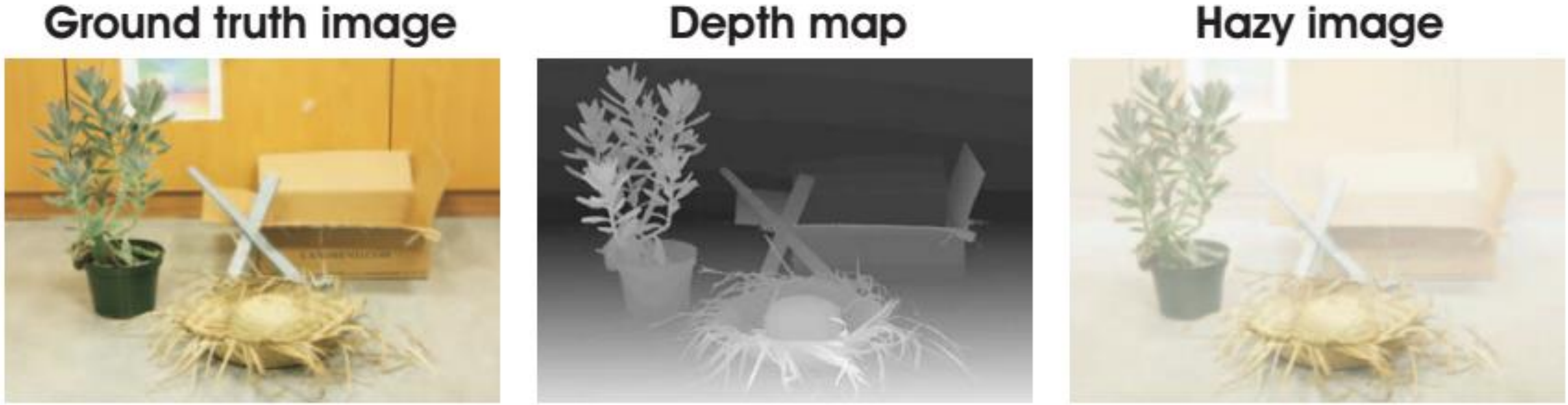


Haze dataset



Haze dataset

- **D-Hazy:**
  - **The dataset that contains 1400+ pairs of images with ground truth reference images and hazy images of the same scene.**



"D-HAZY : A dataset to evaluate quantitatively dehazing algorithms"C. Ancuti, C.Ancuti,C.D.Vleeschouwer. ICIP. 2016

- **D-Hazy:**
  - **The dataset is built on the Middelbury[1] and NYU[2] Depth datasets that privide images of various scenes and their corresponding depth maps.**

[1]http://vision.middlebury.edu/stereo/data/scenes2014/
[2]http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html

- **D-Hazy:**
  - $T(x)=e^{[\beta d(x)]}$
  - The transmission map is estimated based on equation using the depth **d** and the medium attenuation coefficient **β**.
  - $\beta = 1$
  - $A \quad = [1,1,1]$

Dehaze Methods

- **Dark Channel Prior[1]**
- **Color Attenuation Prior[2]**
- **DehazeNet[3]**
- **MSCNN[4]**

1.K. He, J.Sun, and X. Tang,"Single Image Haze Removal Using Dark Channel Prior". PAMI. 2011
2.Q. Zhu, J. Mai, and L. Shao, "A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior". TIP. 2015
3.B. Cai, X. Xu, K Jia, C. Qing, and D. Tao,"DehazeNet:An End-to-End System for Single Image Haze Removal". arXiv. 2016
4.W. Ren, S. Liu, H. Zhang, J. Pan, "Single Image Dehazing via Multi-Scale Convolutional Neural Network" ECCV. 2016

Dehaze Methods

- **Dark Channel Prior**
- **Color Attenuation Prior**
- **DehazeNet**
- **MSCNN**

Single Image Haze Removal Using Dark Channel Prior

- **Dark channel prior:**
  - **It is a kind of statistics of the haze-free outdoor images.**
  - **It is based on a key observation :**
    - **Most local patches in haze-free outdoor images contain some pixels which have very low intensities in at least one color channel.**

Dark Channel

- min (rgb, local patch)
  - min (r, g, b)

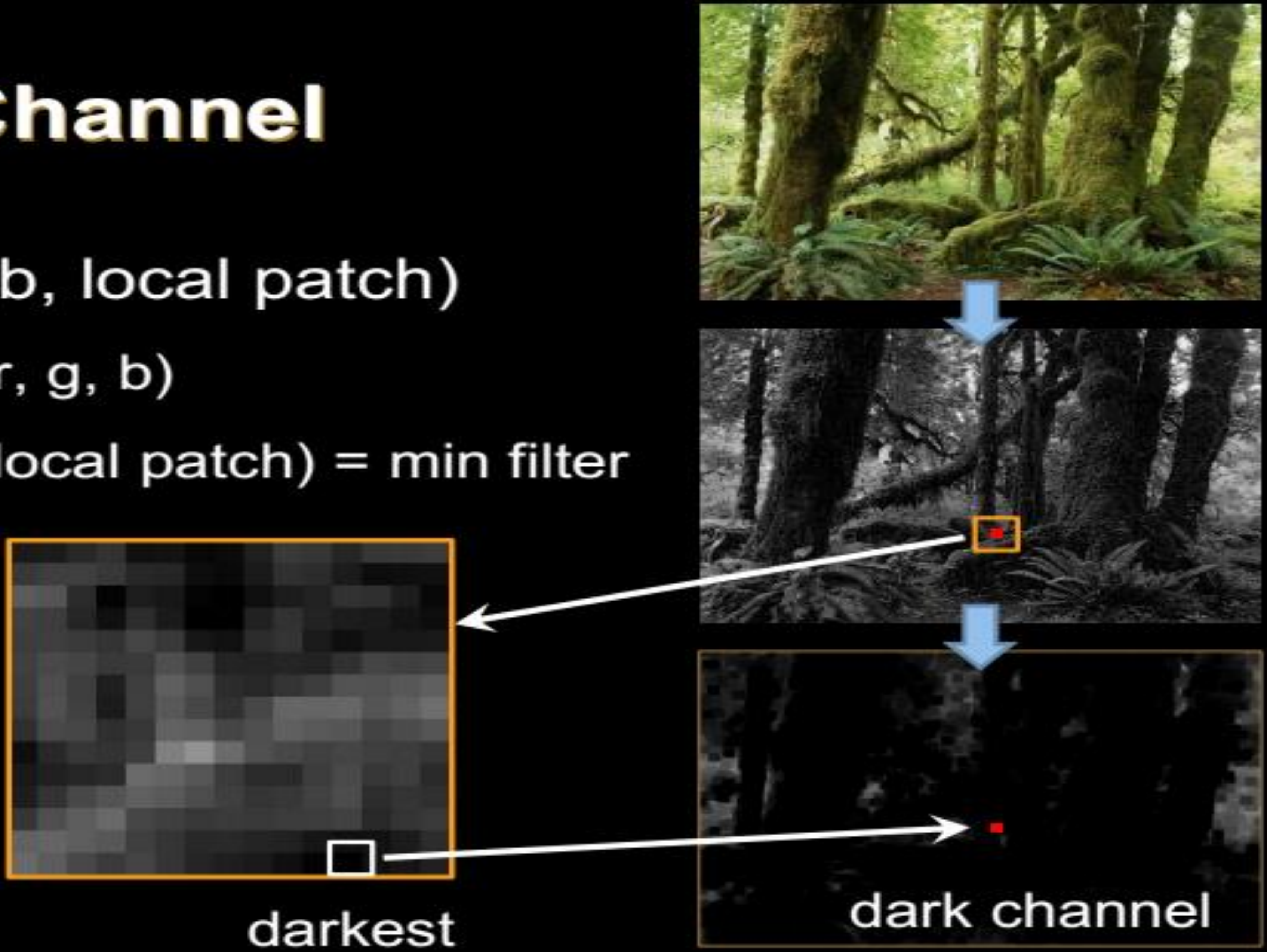
min (r, g, b)

*This slide is from Kaiming He’s talking slides in CVPR’09.

Dark Channel

- min (rgb, local patch)
  - min (r, g, b)
  - min (local patch) = min filter

15 x15



*This slide is from Kaiming He’s talking slides in CVPR’09.

Dark Channel

- min (rgb, local patch)
  - min (r, g, b)
  - min (local patch) = min filter

$$\boxed{J^{dark}(\mathbf{x}) = \min_{y \in \Omega(\mathbf{x})} \left(\min_{c \in \{r,g,b\}} J^c(\mathbf{y}) \right)}$$

  - J^c: color channel of $\mathbf{J}$
  - J^{dark}: dark channel of $\mathbf{J}$

*This slide is from Kaiming He’s talking slides in CVPR’09.

Dark Channel

- min (rgb, local patch)
  - min (r, g, b)
  - min (local patch) = min filter

$$\boxed{\mathrm{J}^{dark} = \min_{\Omega}\left(\min_{c} \mathrm{J}^{c}\right)}$$

  - J^{c}: color channel of $\mathbf{J}$
  - $\mathrm{J}^{\mathrm{dark}}$: dark channel of $\mathbf{J}$

*This slide is from Kaiming He’s talking slides in CVPR’09.

Single Image Haze Removal Using Dark Channel Prior

- **Dark channel prior:**
  - Image size: 500x500, patch size: 15x15
  - 75% dark channel values = 0
  - 86% dark channel values < 16
  - 90% dark channel values < 25

$$\Longrightarrow \quad J^{dark} \rightarrow 0$$

- So : $\min\limits_{\Omega}\left(\min\limits_{c}\left(J^{c}\right)\right) \rightarrow 0$

Single Image Haze Removal Using Dark Channel Prior

- **Transmission Estimation**

$$I(x) = J(x)t(x)+(1\text{-}t(x))A$$

$$\Rightarrow \min_{y\in\Omega(x)}\left(I^c(y)\right)=\tilde{t}(x)\min_{y\in\Omega(x)}\left(J^c(y)\right)+\left(1-\tilde{t}(x)\right)A^c$$

$$\Rightarrow \min_{y\in\Omega(x)}\left(\frac{I^c(y)}{A^c}\right)=\tilde{t}(x)\min_{y\in\Omega(x)}\left(\frac{J^c(y)}{A^c}\right)+\left(1-\tilde{t}(x)\right)$$

$$\min_{y\in\Omega(x)}\left(\min_c\left(\frac{I^c(y)}{A^c}\right)\right)=\tilde{t}(x)\min_{y\in\Omega(x)}\left(\min_c\left(\frac{J^c(y)}{A^c}\right)\right)+(1-\tilde{t}(x))$$

• Dark Channel Prior: Transmission Estimation

$$\min_{y\in\Omega(x)}\left(\min_{c}\left(\frac{I^{c}(y)}{A^{c}}\right)\right)=\tilde{t}(x)\boxed{\min_{y\in\Omega(x)}\left(\min_{c}\left(\frac{J^{c}(y)}{A^{c}}\right)\right)}+(1-\tilde{t}(x))$$

$$J^{dark}(x)=\min_{y\in\Omega(x)}\left(\min_{c}\left(J^{c}(y)\right)\right)=0, and, A^{c}>0$$

$$\boxed{\min_{y\in\Omega(x)}\left(\min_{c}\left(\frac{J^{c}(y)}{A^{c}}\right)\right)}=0$$

$$\min_{y\in\Omega(x)}\left(\min_{c}\left(\frac{I^{c}(y)}{A^{c}}\right)\right)=1-\tilde{t}(x)$$

$$\tilde{t}(x)=1-\omega\min_{y\in\Omega(x)}\left(\min_{c}\left(\frac{I^{c}(y)}{A^{c}}\right)\right)$$

Single Image Haze Removal Using Dark Channel Prior

He shows that the **transmission** can be estimated by calculating:

$$\tilde{t}(x) = 1 - \omega \min_{y \in \Omega(x)} \left(\min_{c} \left(\frac{I^c(y)}{A^c} \right) \right)$$

$$\omega = 0.95$$

Single Image Haze Removal Using Dark Channel Prior

- **Atmospheric Light (A) Estimation:**
  - **Airlight** is estimated by picking up the pixels of the image corresponding to the 0.1% brightest pixels in the dark channel.
  - And then choosing the pixels with maximum intensity.

Single Image Haze Removal Using Dark Channel Prior

$$I(x) = J(x)t(x) + A(1 - t(x))$$

$$\Rightarrow J(x) = \frac{I(x) - A(1 - t(x))}{t(x)}$$

$$\Rightarrow J(x) = \frac{I(x) - A + At(x)}{t(x)}$$

$$\Rightarrow J(x) = \frac{I(x) - A}{t(x)} + A \quad \Longrightarrow \quad J(x) = \frac{I(x) - A}{\max(t(x), t_0)} + A$$

Single Image Haze Removal Using Dark Channel Prior

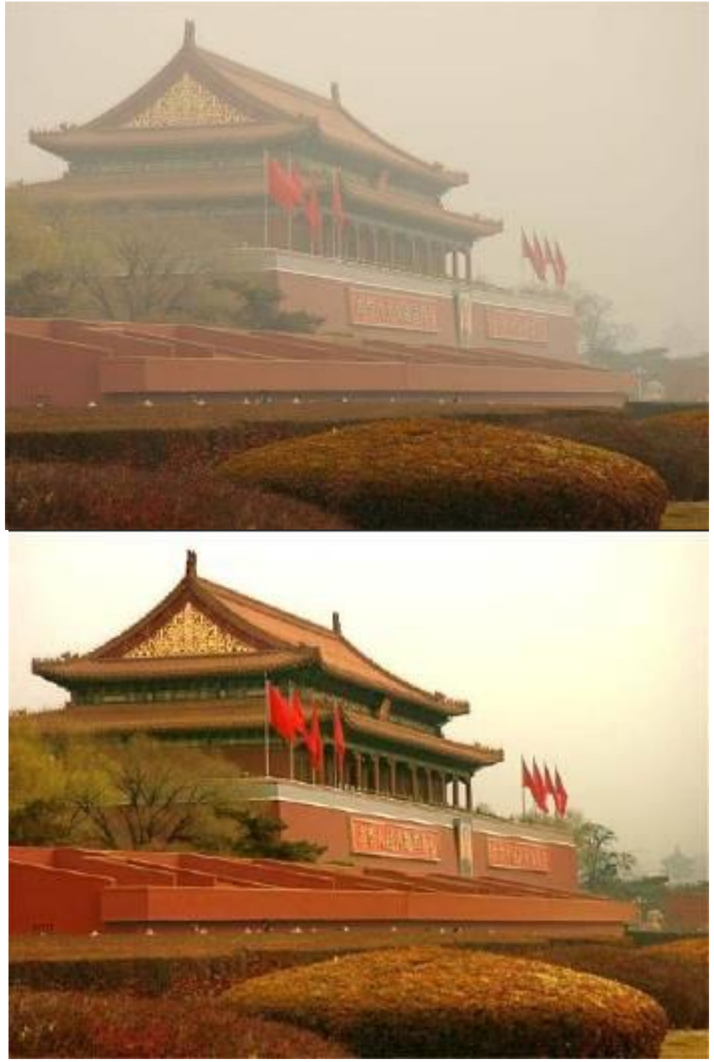
$$J(x) = \frac{I(x) - A}{\max(t(x), t_0)} + A \quad \longrightarrow \quad t_0 = 0.1$$

Single Image Haze Removal Using Dark Channel Prior

- Airlight and transmission are sufficient to invert the model and recover the original radiance of the scene.

$$J(x) = \frac{I(x) - A}{\max\left(t(x), t_0\right)} + A$$

Single Image Haze Removal Using Dark Channel Prior



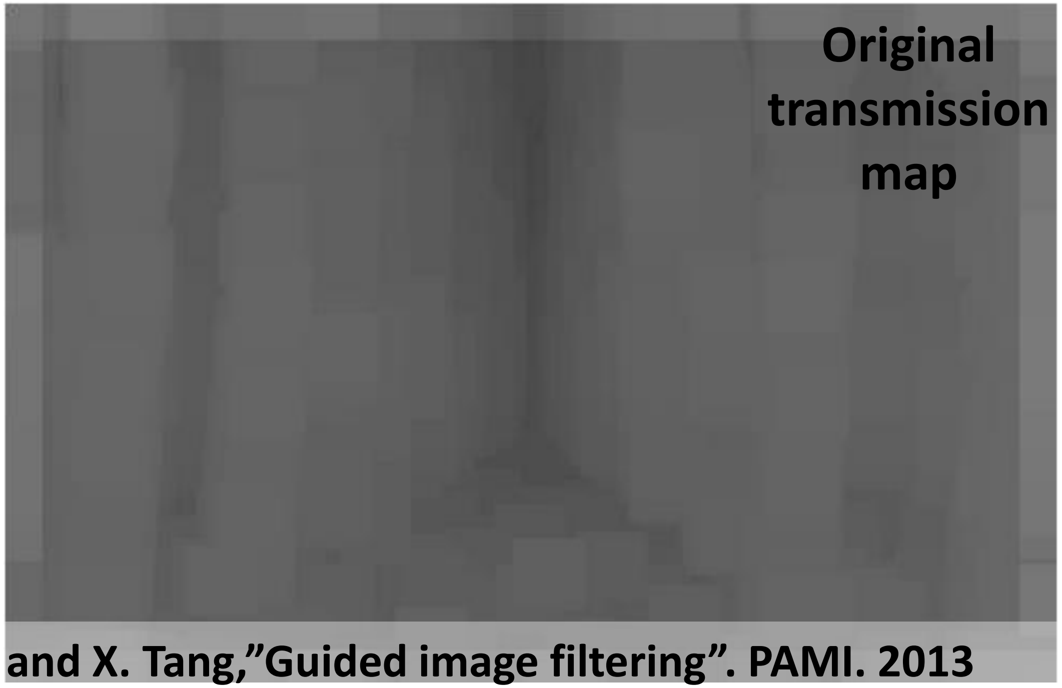
Single Image Haze Removal Using Dark Channel Prior

haze **dehaze**

Single Image Haze Removal Using Dark Channel Prior

haze

dehaze

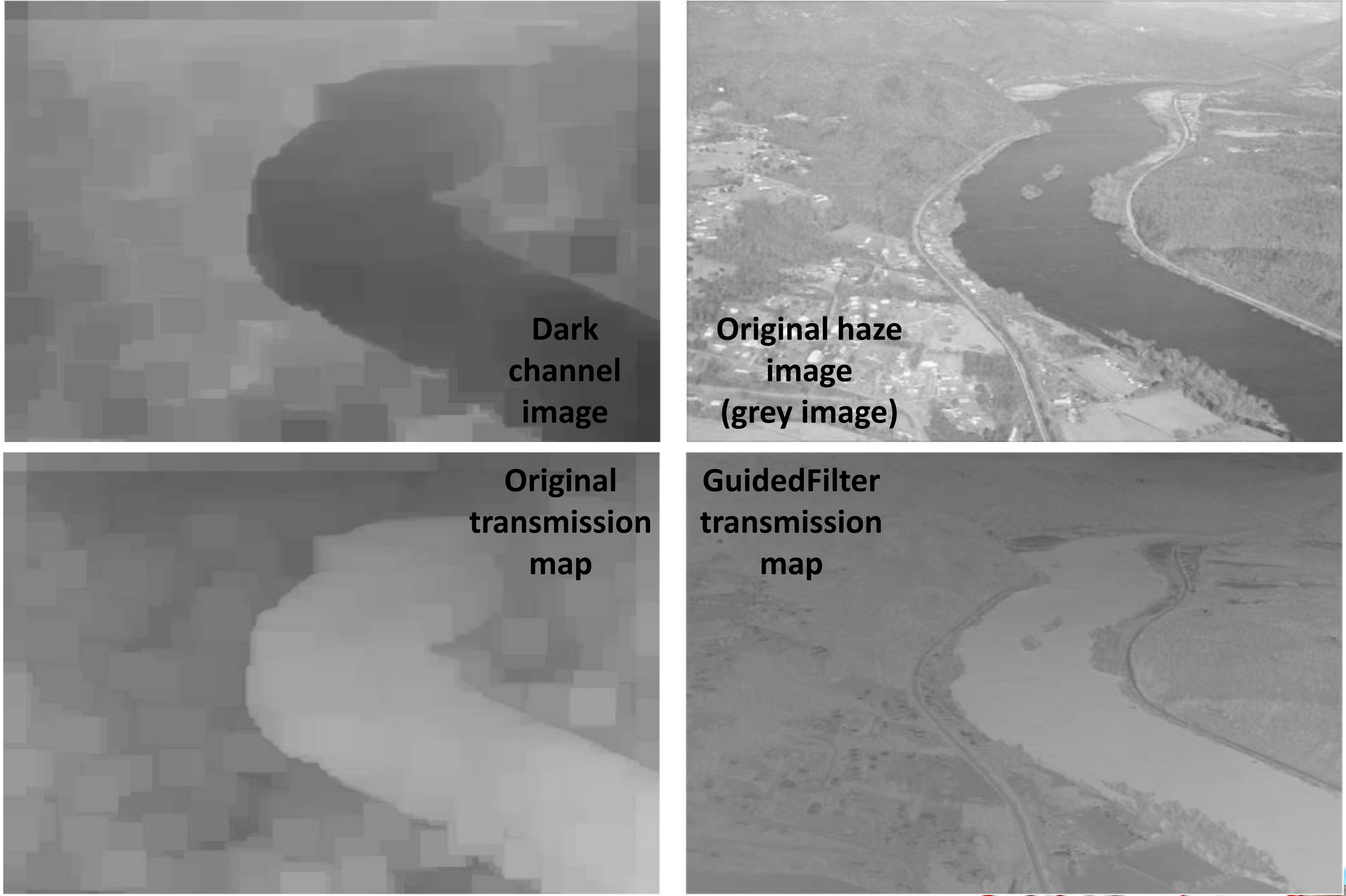


K. He, J.Sun, and X. Tang,"Guided image filtering". PAMI. 2013

Single Image Haze Removal Using Dark Channel Prior

haze

dehaze



K. He, J.Sun, and X. Tang,"Guided image filtering". PAMI. 2013

Single Image Haze Removal Using Dark Channel Prior

Original Dark Channel Prior

Original

Dark Channel Prior

Single Image Haze Removal Using Dark Channel Prior

- **Limitation:**

It is known what are the consequences of a bad estimate for the transmission-------->

Haze is not completely removed, or it is removed where there is no haze (overboost contrast)

Single Image Haze Removal Using Dark Channel Prior

- **Limitation:**

When the scene objects are similar to the atmospheric light and no shadow is cast on them, the dark channel prior is invalid.

The method will underestimate the transmission for these objects, such an the white marble in Figure.

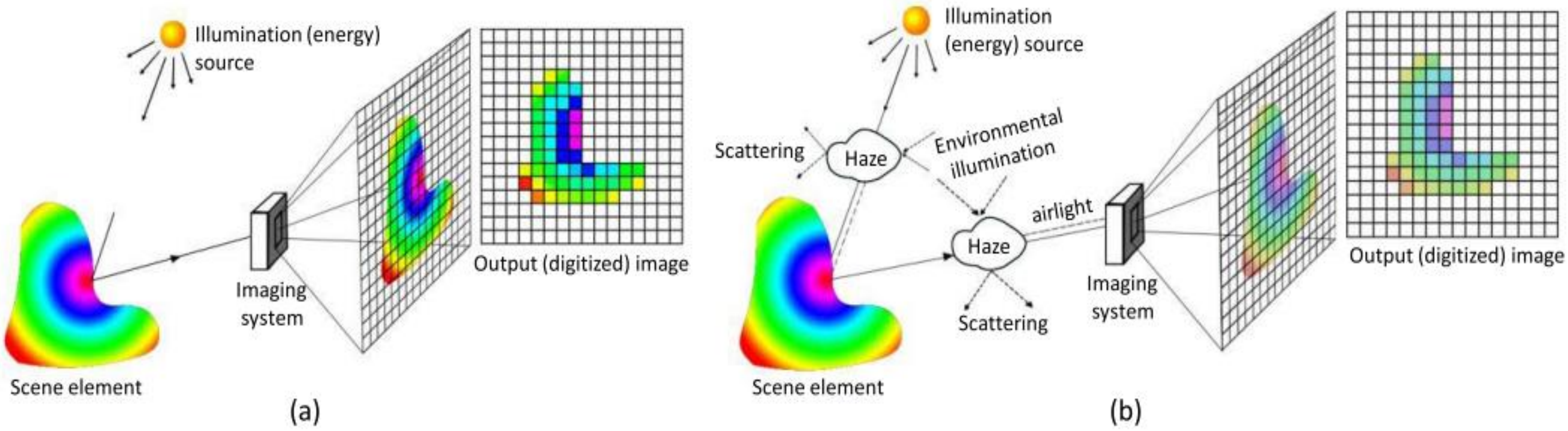
Figure 13. Failure case. Left: input image. Middle: our result. Right: our transmission map. The transmission of the marble is underestimated.

Dehaze Methods

- Dark Channel Prior
- Color Attenuation Prior
- DehazeNet
- MSCNN

A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior

- **Principle:**
  - **The atmospheric scattering model**



How to dehaze

- **Principle:**
  - **The atmospheric scattering model**

$$I(x) = J(x)t(x)+A(1-t(x))$$

    - **I(x)---->the observed haze image**
    - **J(x)---->the real scene to be recovered**
    - **t(x)---->the medium transmission**
    - **A---->the global atmospheric light(Airlight)**
    - **x---->index pixels in the observed haze image**

How to dehaze

- **Principle:**
  - **The atmospheric scattering model**

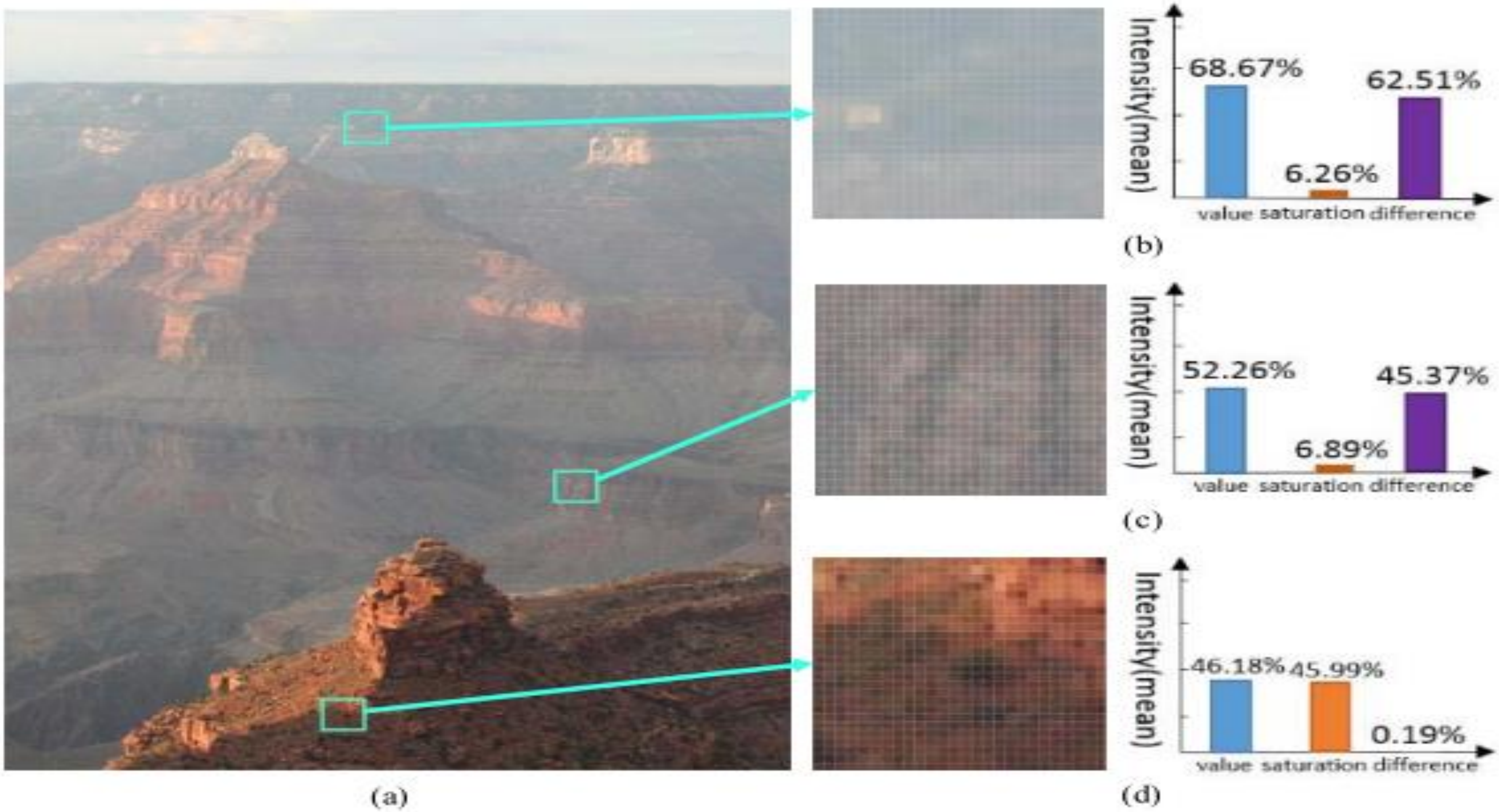
$$I(x) = J(x)t(x)+A(1-t(x))$$

$$t(x) = e^{-\beta d(x)}$$

    - **d(x)---->the distance from the scene point to the camera (depth map)**
    - **β---->the scattering cofficient of the atmosphere**

When the I(x) = A, d(x)--->far away

Color Attenuation Prior



A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior

- **Scene depth recovery**

$$d(x) \propto c(x) \propto v(x) - s(x)$$

d : the scene depth
c : the concentration(浓度) of the haze
v : the brightness(亮度) of the scene
s : the saturation(饱和度)

A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior

- **The Linear Model Definition**

$$d(x) = \theta_0 + \theta_1 v(x) + \theta_2 s(x) + \varepsilon(x)$$

$$\varepsilon(x) \sim N(0, \sigma^2)$$

d : the scene depth

c : the concentration of the haze

v : the brightness of the scene

s : the saturation

$\theta_0 \theta_1 \theta_2$： unknown linear coefficients

ε(x): random variable representing the random error of the model

A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior

$$d(x)-\theta_0-\theta_1 v(x)-\theta_2 s(x)\sim N(0,\sigma^2)$$

$$d(x)\sim N(\theta_0+\theta_1 v(x)+\theta_2 s(x),\sigma^2)$$

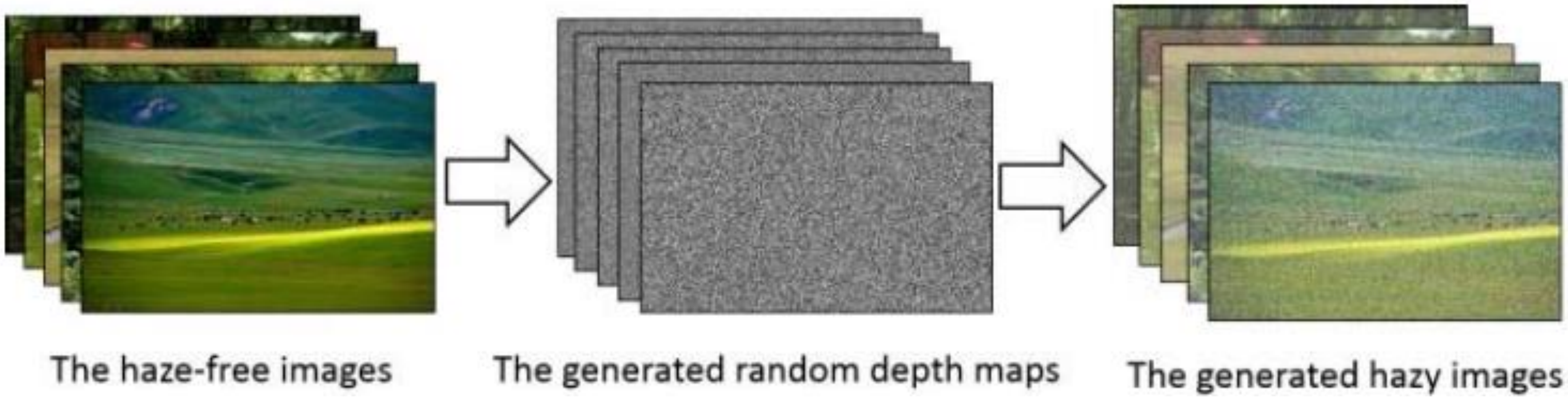
- Scene depth recovery
  - 1.Synthetic dataset
  - 2.Maximum likelihood estimate(最大似然估计)

A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior

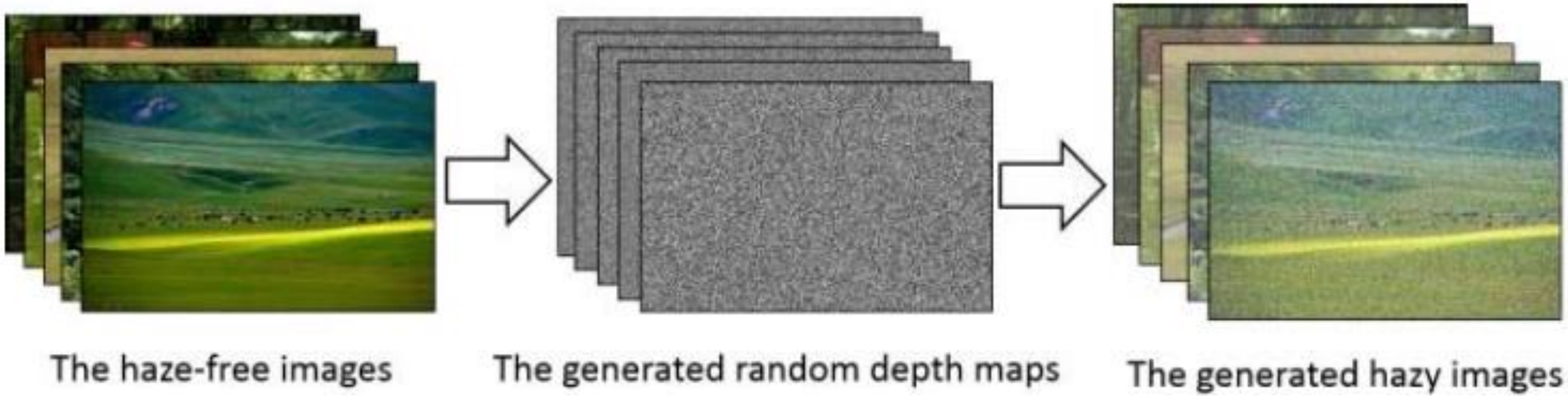
- Scene depth recovery
  - 1.Synthetic dataset
  - 2.Maximum likelihood estimate(最大似然估计)

A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior

- Scene depth recovery
  - 1.Synthetic dataset
  - Clear image + Random depth maps + Random global atmospheric light = Sample haze images



A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior



$$I(x) = J(x)t(x)+A(1-t(x))$$

$$t(x) = e^{-\beta d(x)}$$

$\beta = 1$

$d(x)$ = Random depth maps

A = Random global atmospheric light

A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior

- Scene depth recovery
  - 1.Synthetic dataset
  - 2.Maximum likelihood estimate(最大似然估计)

$$L = \prod_{i=1}^{n} p(d(x_i) \mid x_i, \theta_0, \theta_1, \theta_2, \sigma^2)$$

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(dg_i - (\theta_0 + \theta_1 v(x_i) + \theta_2 s(x_i)))^2$$

$$\begin{cases} \dfrac{\partial \ln L}{\partial \theta_0} = \dfrac{1}{\sigma^2}\sum_{i=1}^{n}(dg_i - (\theta_0 + \theta_1 v(x_i) + \theta_2 s(x_i))) \\ \dfrac{\partial \ln L}{\partial \theta_1} = \dfrac{1}{\sigma^2}\sum_{i=1}^{n} v(x_i)(dg_i - (\theta_0 + \theta_1 v(x_i) + \theta_2 s(x_i))) \\ \dfrac{\partial \ln L}{\partial \theta_2} = \dfrac{1}{\sigma^2}\sum_{i=1}^{n} s(x_i)(dg_i - (\theta_0 + \theta_1 v(x_i) + \theta_2 s(x_i))) \end{cases}$$

A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior

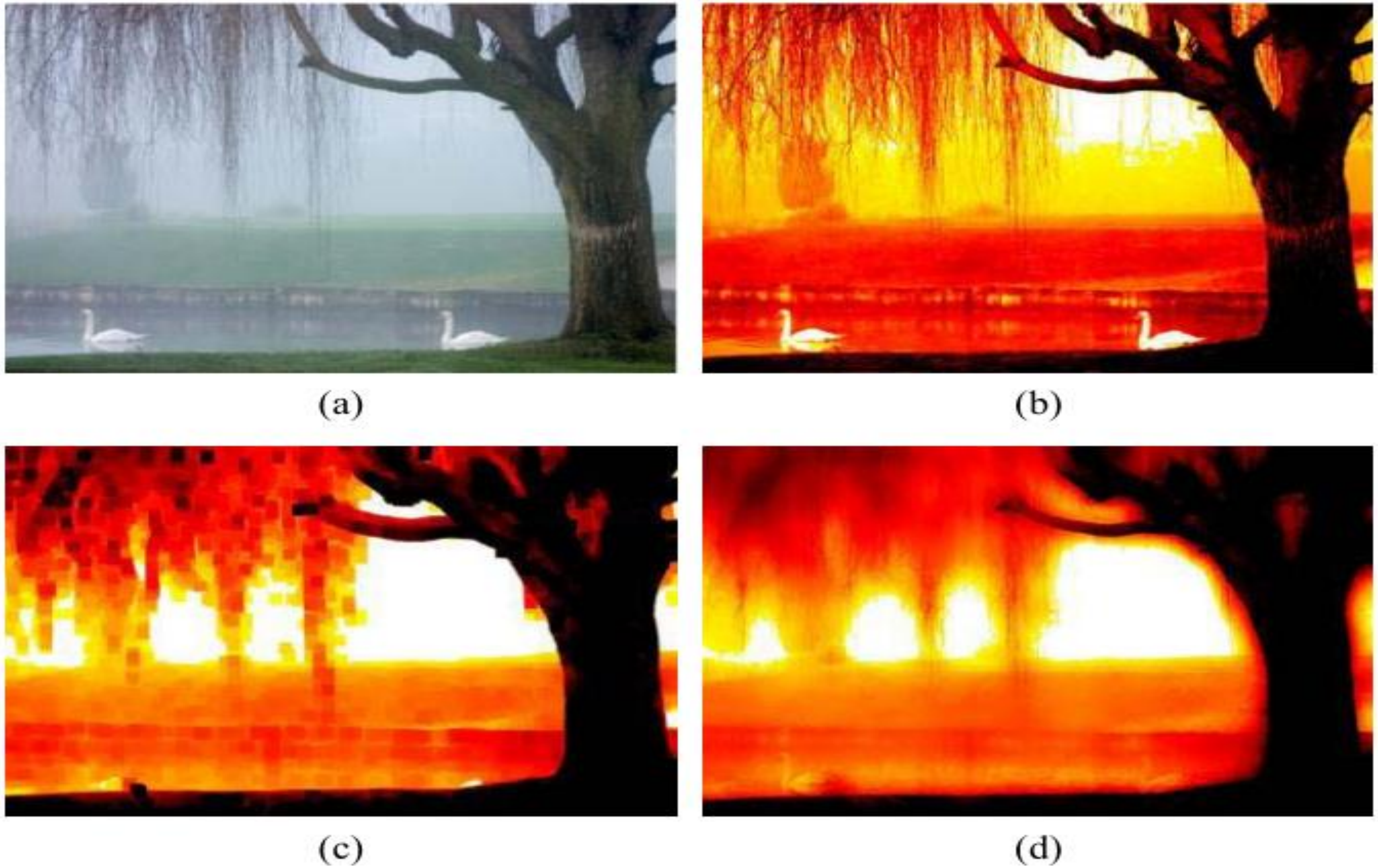
- Scene depth recovery
  - 1.Synthetic dataset
  - 2.Maximum likelihood estimate(最大似然估计)
    - 500 training samples
    - 120 million pixel points

θ_0= 0.121779

θ_1= 0.959710

θ_2=-0.780245

σ = 0.041337



Refinement of the depth map

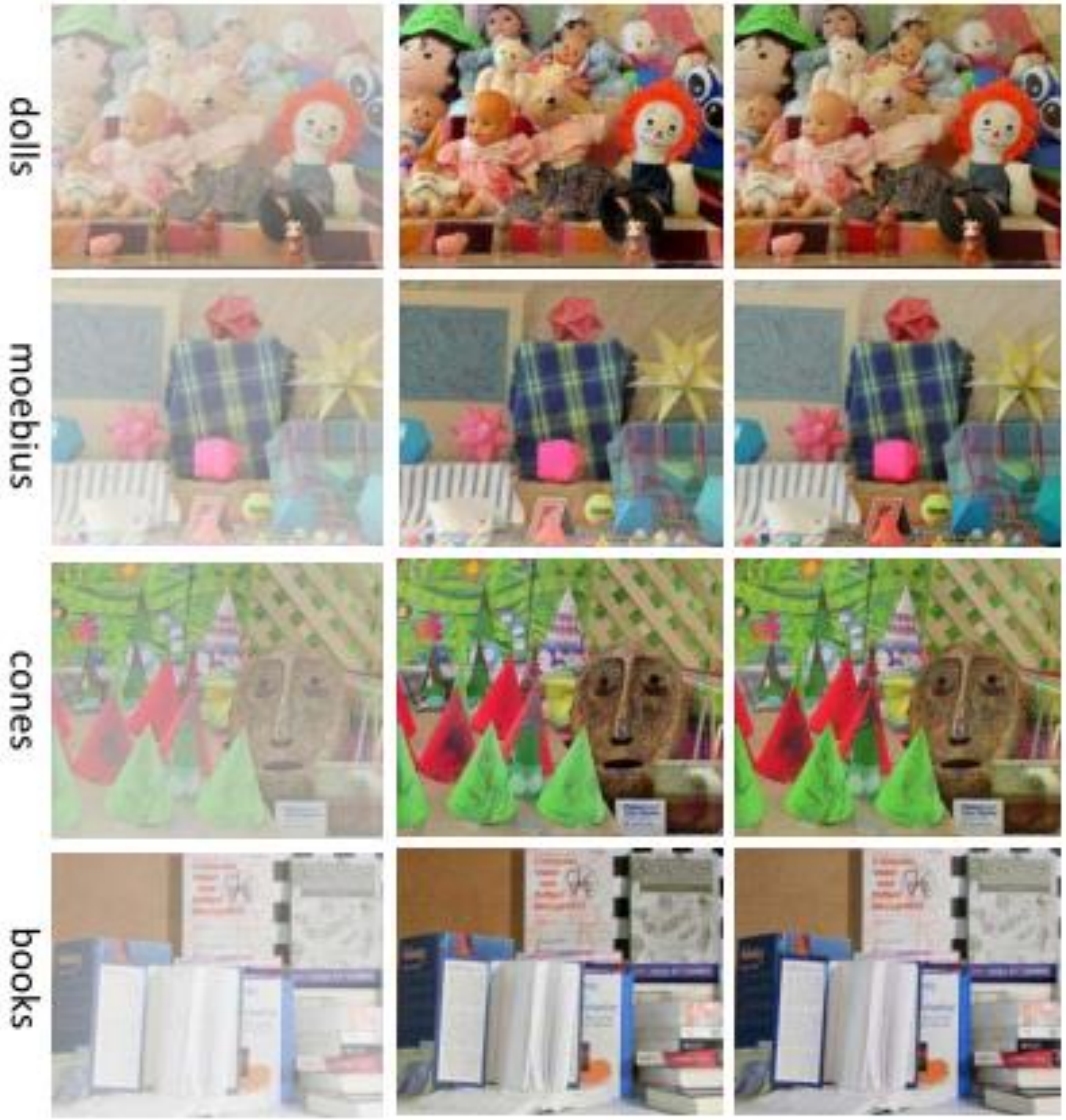
(a) The hazy image **(b) The raw depth**

(c) The depth map with scale r = 15 **(d) The refined depth map**

A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior

(a) (b) (c)

- t(x)---->[0.1,0.9]
- I(x)=J(x)t(x)+A(1-t(x)) ---------> $J(x)=\dfrac{I(x)-A}{\min\{\max\{e^{-\beta d(x)},0.1\}0.9\}}+A$
- $t(x)=e^{-\beta d(x)}$
- β=1.0 (usually)



(a) (b) (c)

Results on stereo images where the ground truth solutions are known.

(a)The haze images
(b)The results
(c)Ground truth

Dehaze Methods


- **Dark Channel Prior**
- **Color Attenuation Prior**
- **DehazeNet**
- **MSCNN**

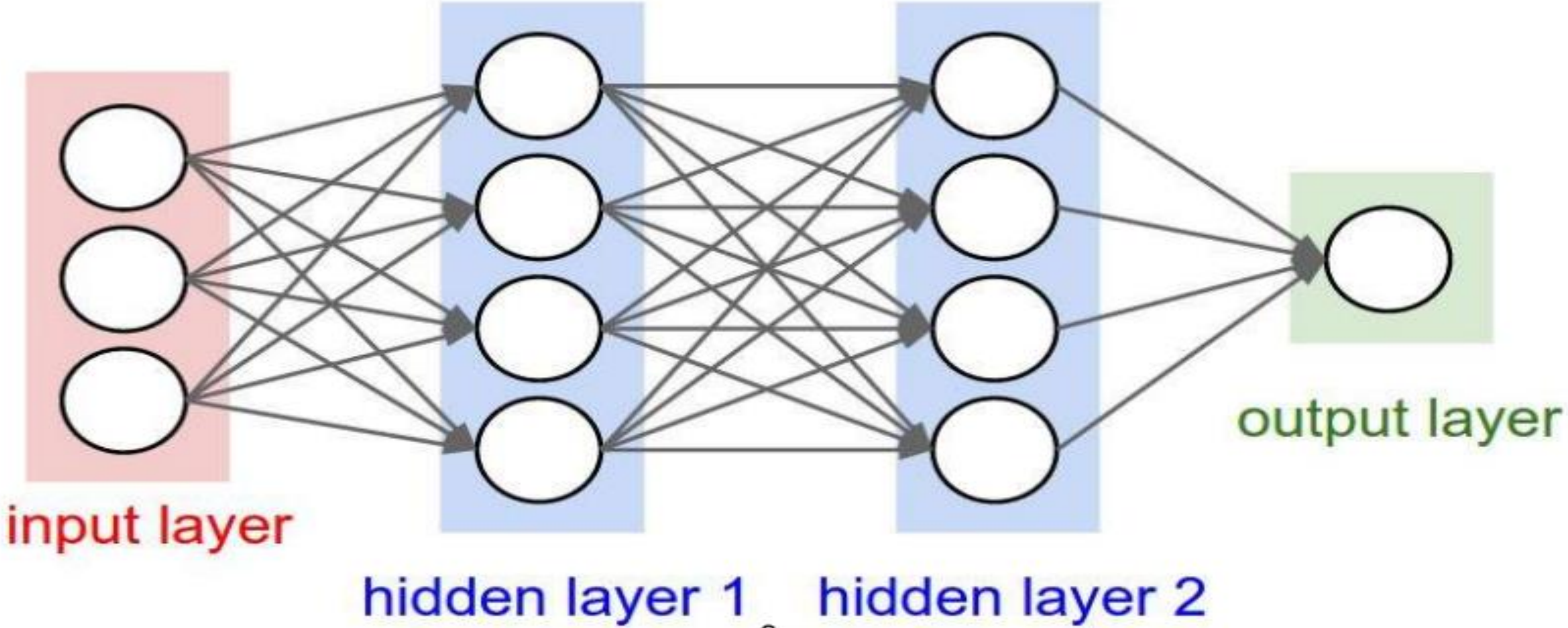
Computer Vision and Convolution Neural Network

➢ **Overview**

- **Hierarchical structure**
- **Visual understanding**
- **Training algorithm**
- **Advantages and disadvantages**

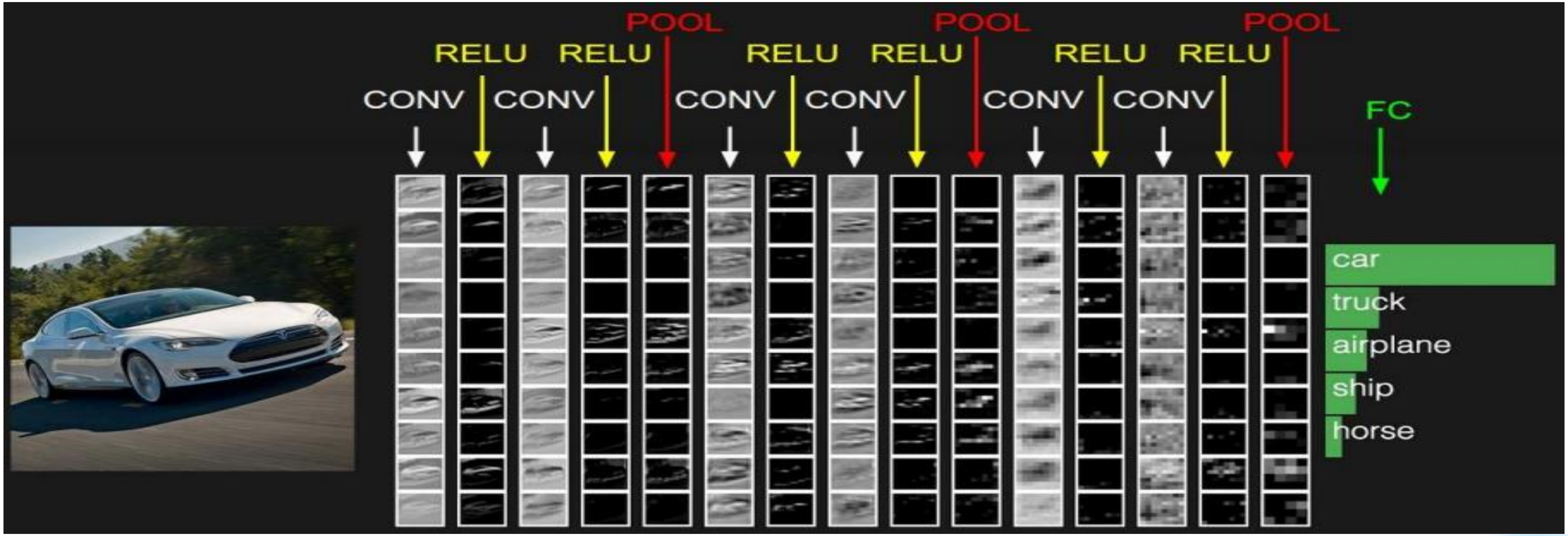
Computer Vision and Convolution Neural Network

- **Muti-layer Perceptron Neural Networks can be applied to computer vision?** YES!
- **Why need Convolutional Neural Networks?**
- **Where are the differences between Convolutional Neural Networks and Muti-layer Perceptron Neural Networks?**



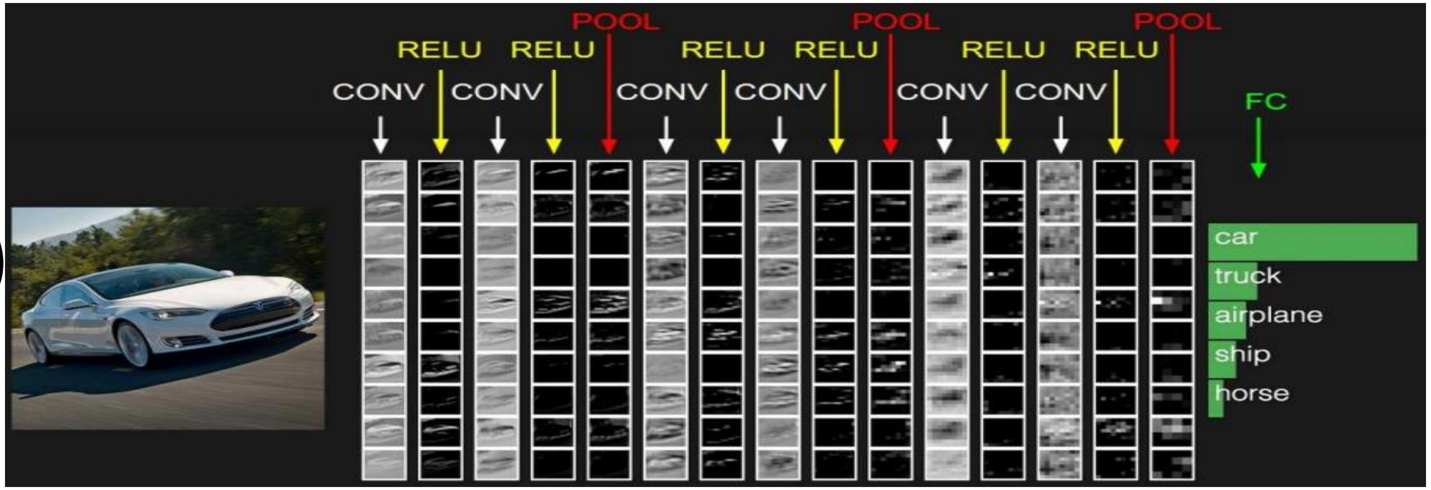
Hierarchical Structure of CNN

- **Keep a hierarchical network structure**
- **Different Hierarchies have different forms (operation) and functions**



Hierarchical Structure of CNN

- ➢Input layer(数据输入层)
- ➢Conv layer(卷积计算层)
- ➢ReLU layer(ReLU激励层)
- ➢Pooling layer(池化层)
- ➢FC layer(全连接层)
- ➢Batch Normalization (maybe)



Input layer

➢Data processing mode

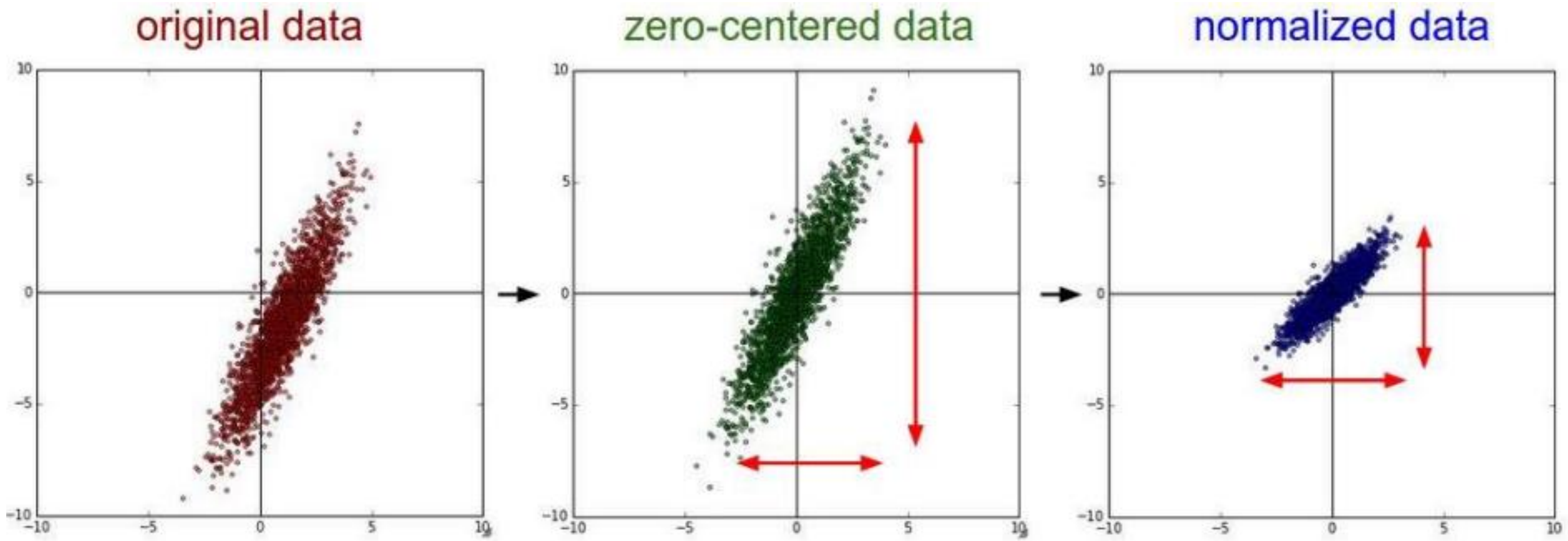
1. **De mean:** Various dimensions of input data to zero
2. **Normalization:** Amplitude normalized to the same range
3. **PCA(Principal Components Analysis):**

   To dimension reduction

   The amplitude(幅度) in data feature axis to normalization

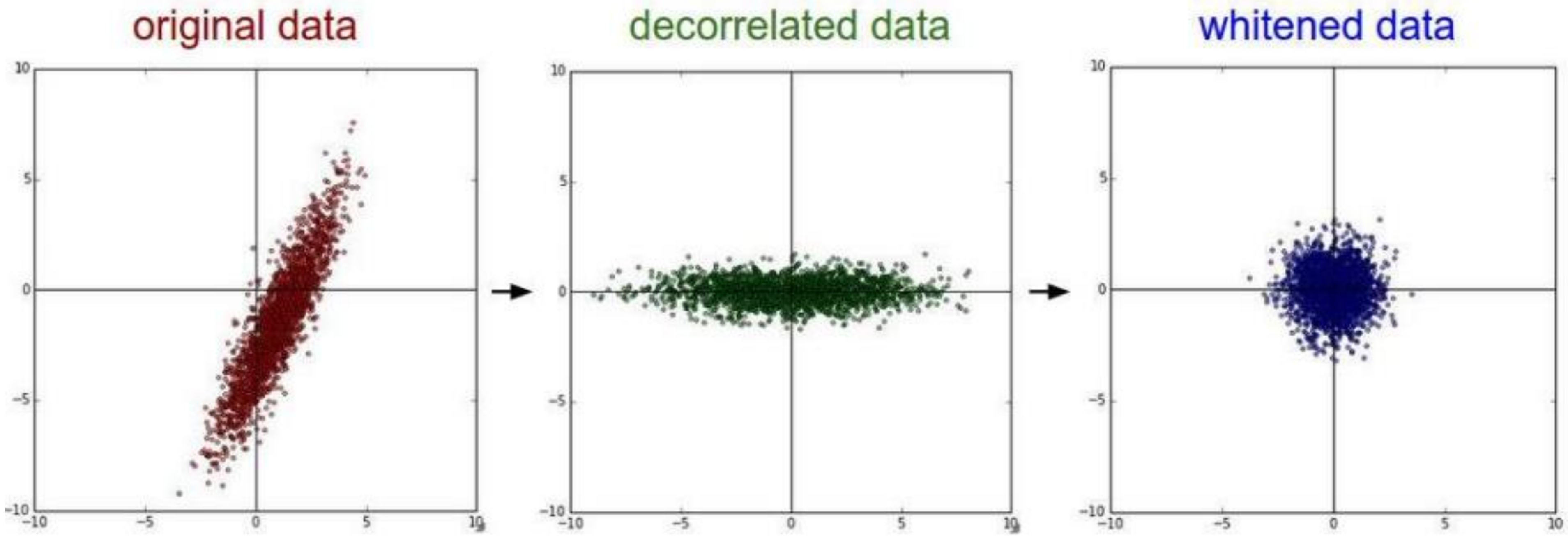
Input layer

- **De mean and normalization**



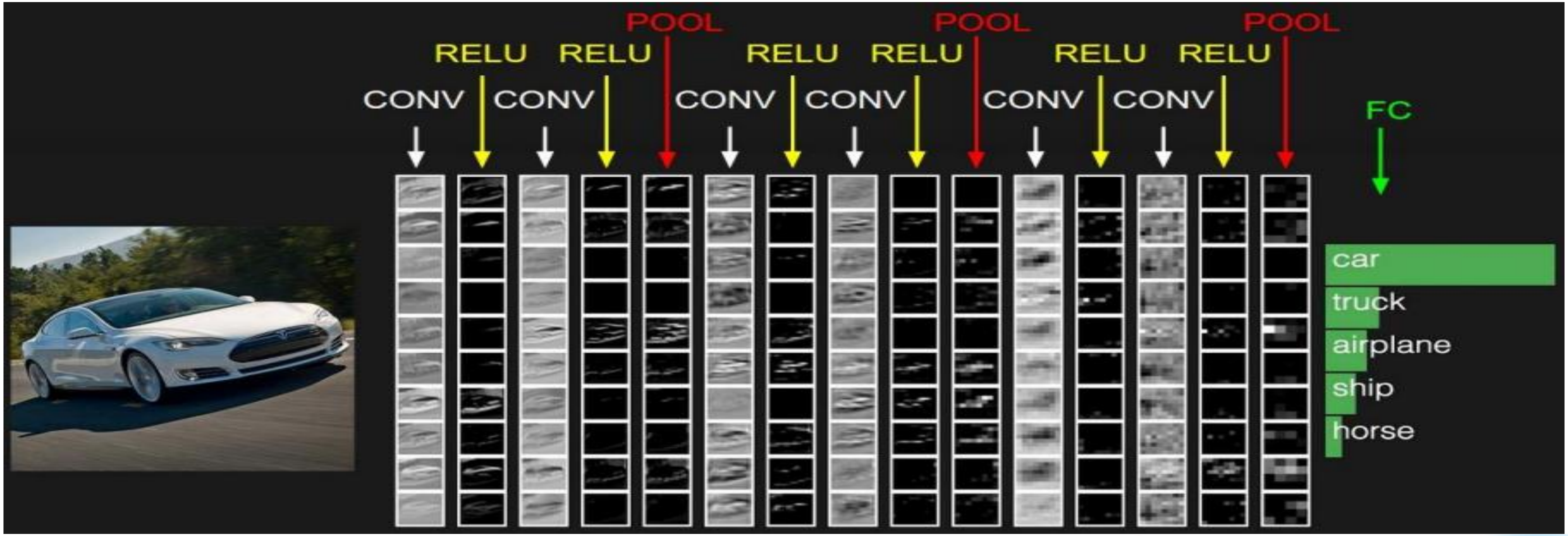
Input layer

- **Decorrelation and PCA(Principal Components Analysis)**



Hierarchical Structure of CNN

- **Keep a hierarchical network structure**
- **Different Hierarchies have different forms (operation) and functions**



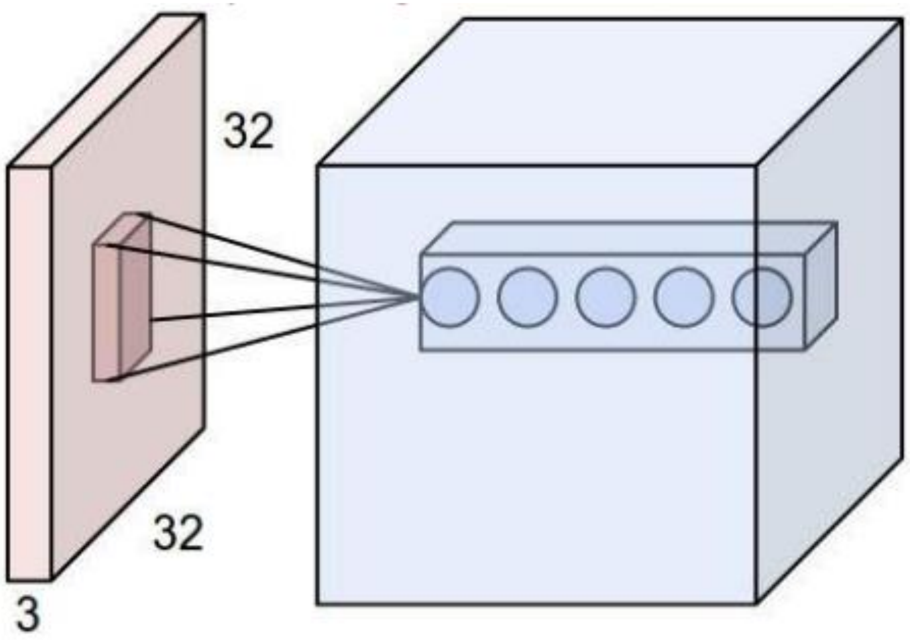
Conv layer

➢ **Local association: Each neuron is seen as a filter**

➢ **Windows(receptive filed) slide, filter can compute the local data**

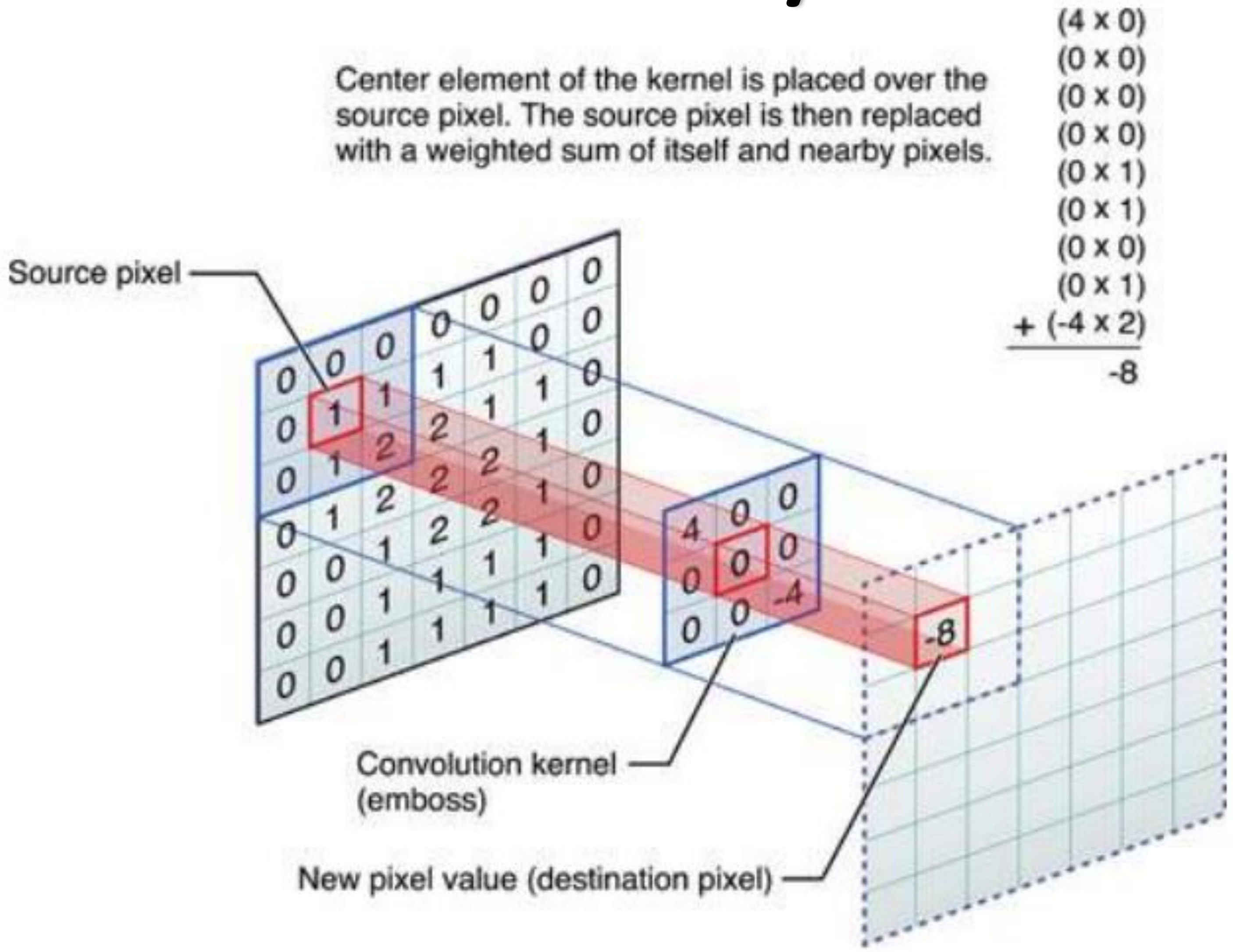
➢ **Concepts:**

- **depth**
- **stride**
- **zero-padding**



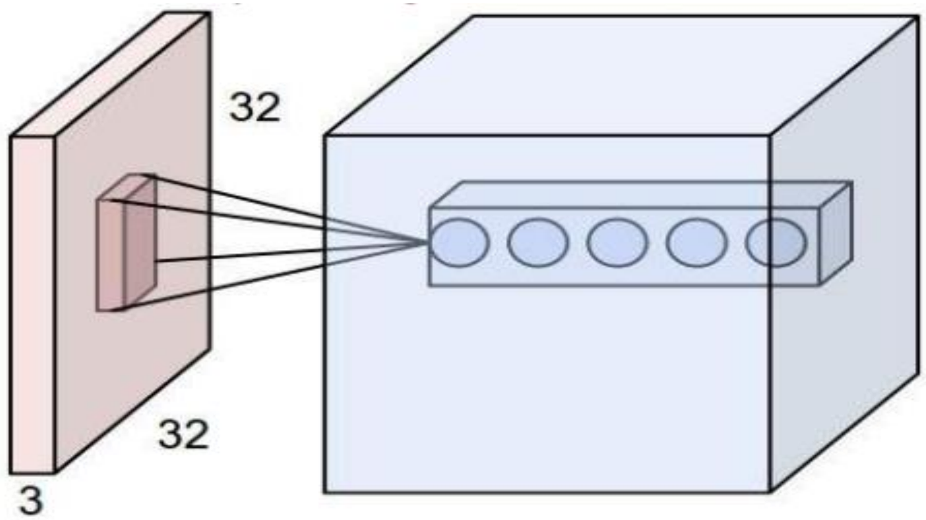
http://blog.csdn.net/loving_forever_/article/details/52389862

Conv layer


Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.
(4 x 0)
(0 x 0)
(0 x 0)
(0 x 0)
(0 x 1)
(0 x 1)
(0 x 0)
(0 x 1)
+ (-4 x 2)
-8
Source pixel
Convolution kernel
(emboss)
New pixel value (destination pixel)
-8

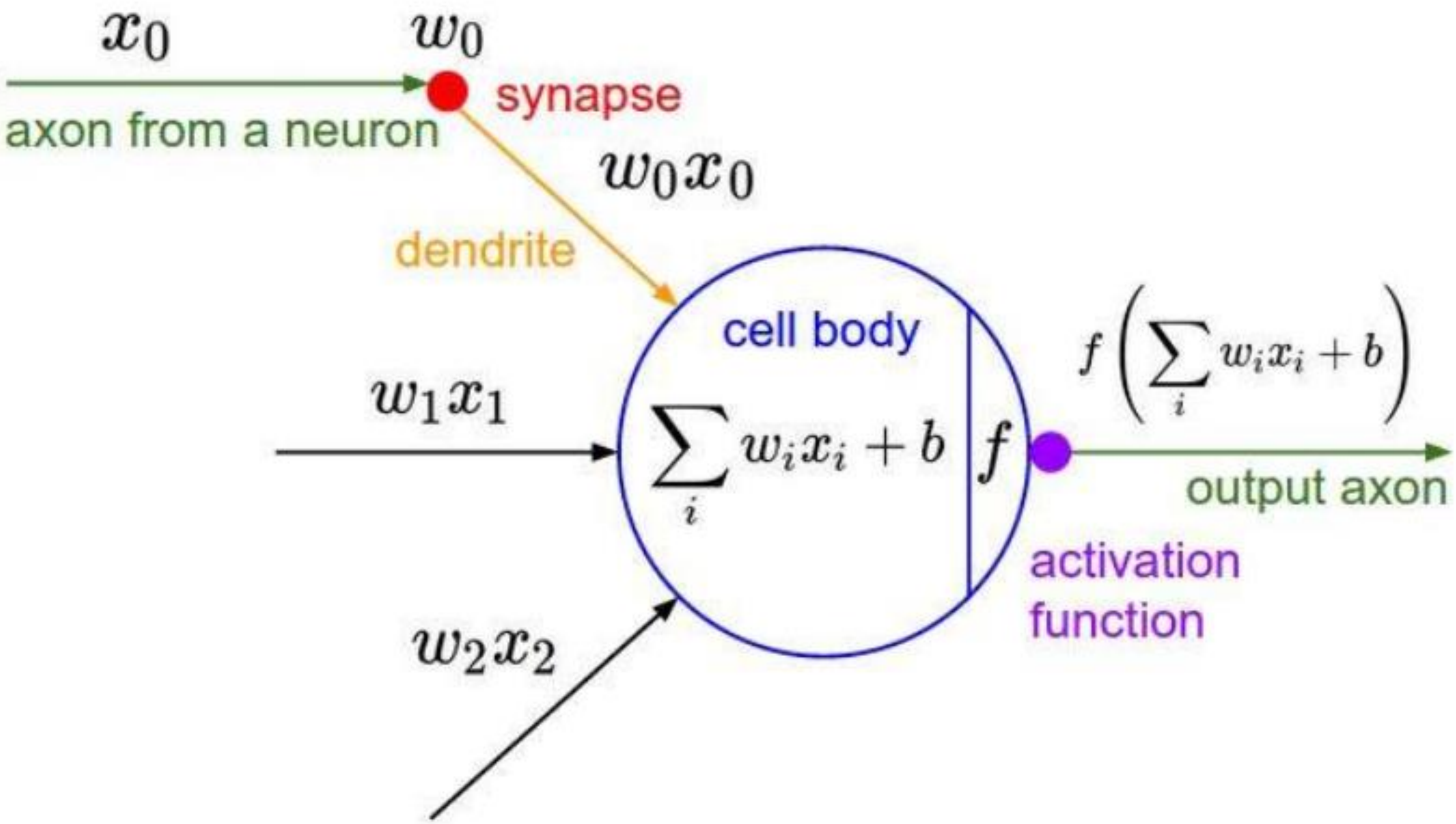
Conv layer

- **Parameter sharing mechanism(参数共享机制)**
  - If the parameters in weight is fixed, it can be seen the template
    - (each neuron only follow one feature)
  - Reduced number of the estimate weights
    - One hundred million===>35 thousands
  - A group of fixed weights and different windows to make convolution



Relu layer

- The results of the output in convolution layers to nonlinear mapping (非线性映射)



Relu layer

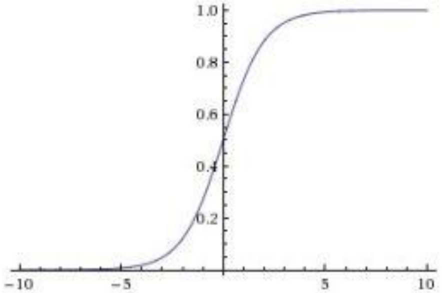
➢The results of the output in convolution layers to nonlinear mapping

- Sigmoid
- Tanh(双曲正切)
- ReLU
- Leaky ReLU
- ELU
- Maxout

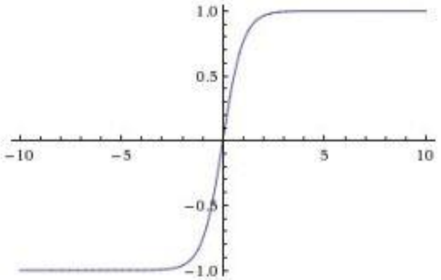
Activation Functions

Sigmoid

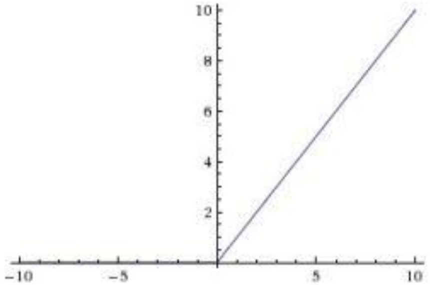
$$\sigma(x) = 1/(1 + e^{-x})$$



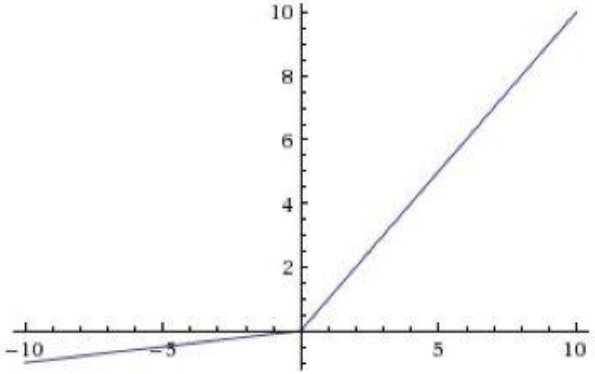
tanh tanh(x)



ReLU max(0,x)

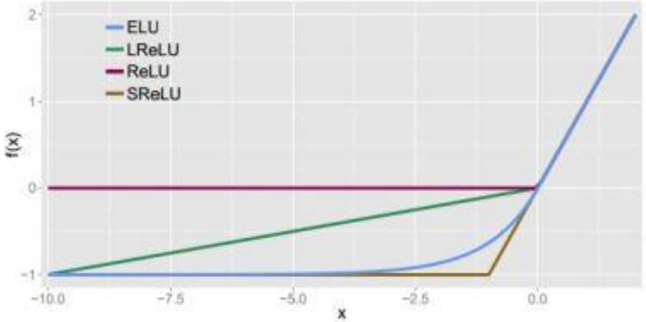


Leaky ReLU
max(0.1x, x)



Maxout $\max(w_1^T x + b_1, w_2^T x + b_2)$

ELU
$$f(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha\,(\exp(x) - 1) & \text{if } x \leq 0 \end{cases}$$



Relu layer

Experience:

1.CNN would better not use Sigmoid.

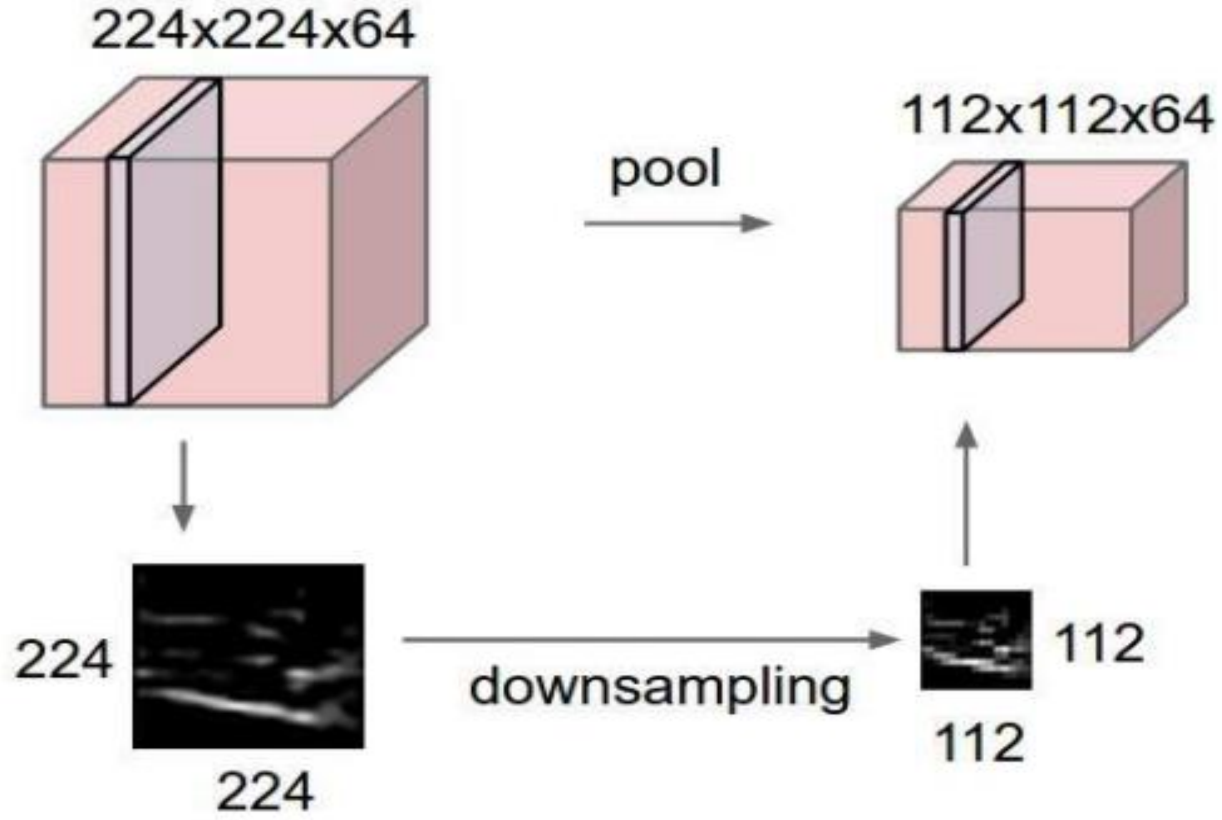
2.You may try ReLU first because it is fast(Be careful).

3.If 2 doesn't work, please try Leaky ReLU or Maxout

4.Little time Tanh can work well, you can also have a try.

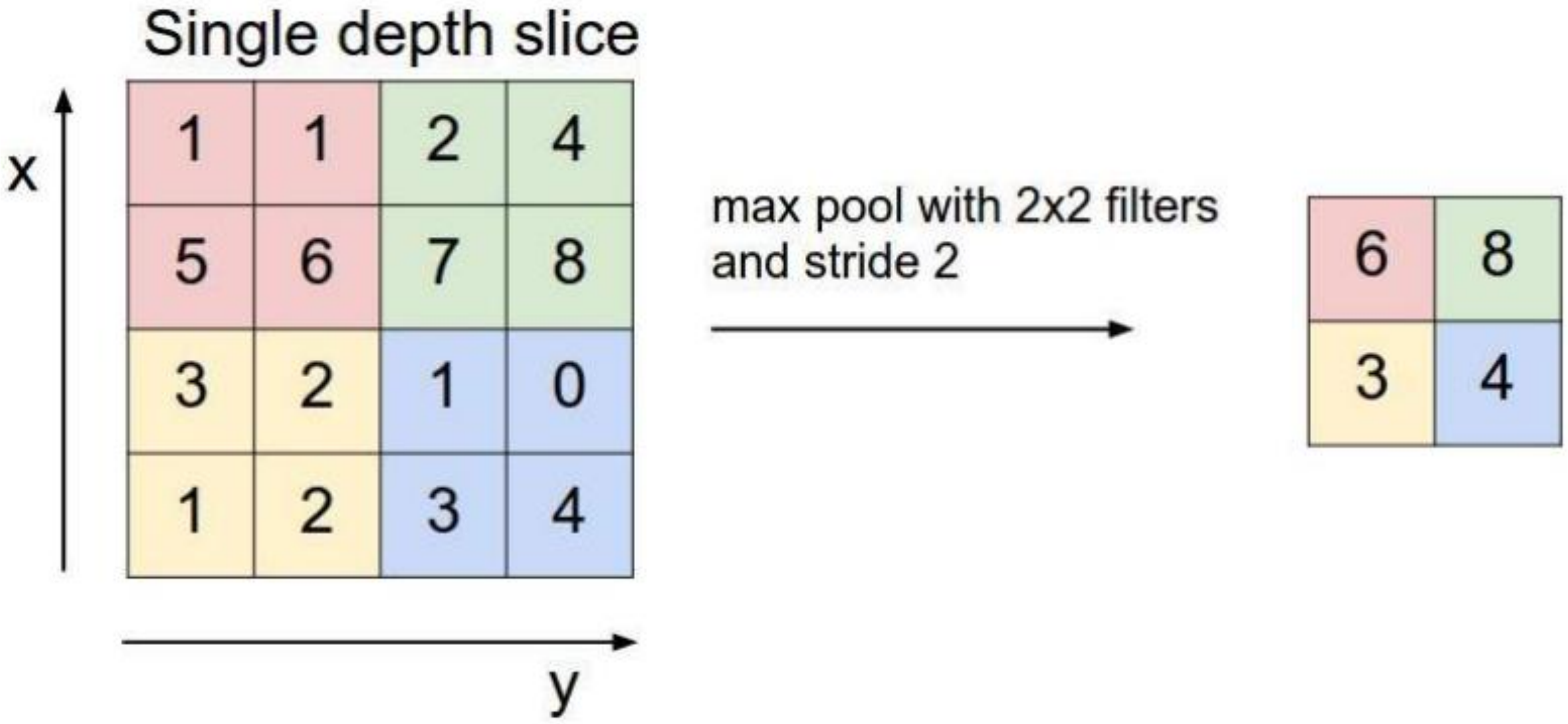
Pooling layer

- Clip in the convolution layers
- Compress data and parameters, reduce overfitting



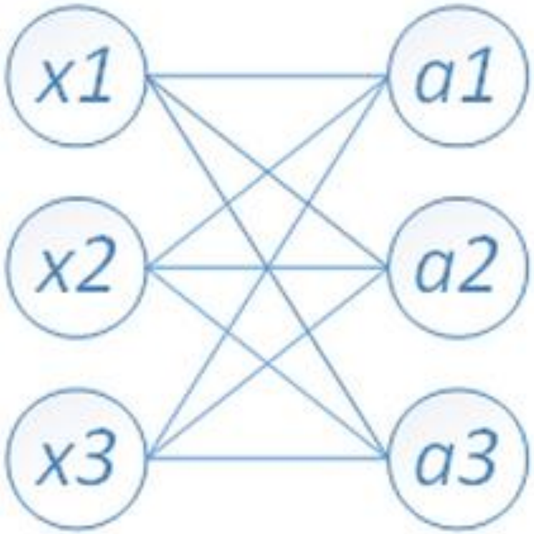
Pooling layer

- Max pooling
- Average pooling



FC layer

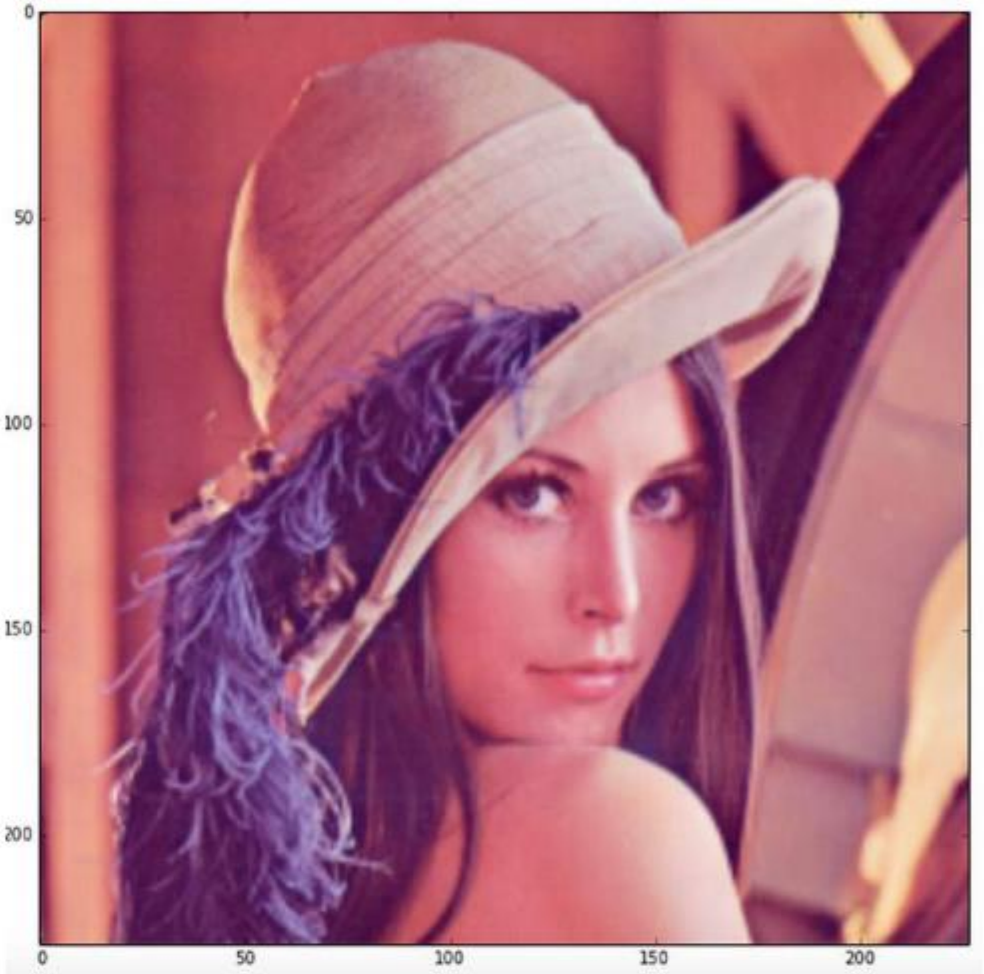
- Between two layers, all neurons have the connection with weights
- The FC layers usually in the end of the convolutional neural network



The Structure in the CNN

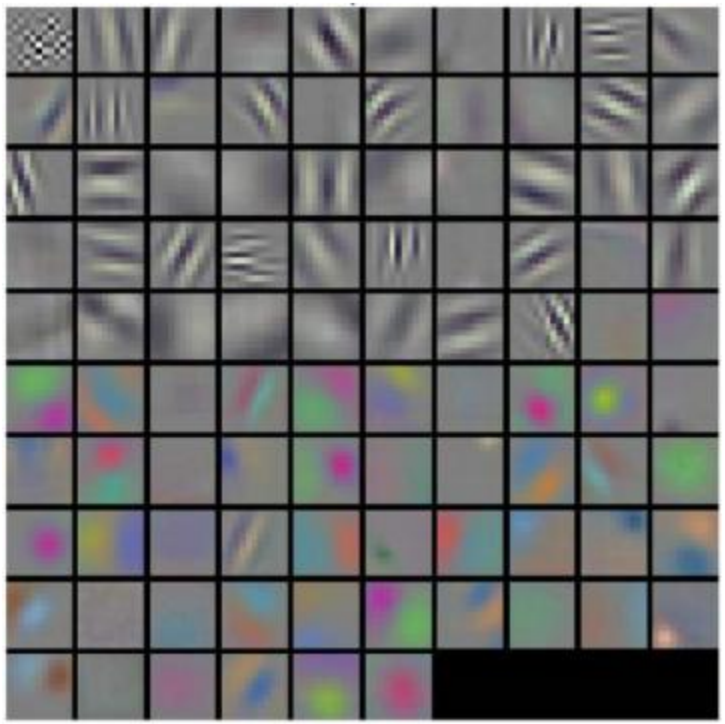
- Input
- [[Conv->Relu]*N->Pool]*M
- [FC->Relu]*K
- FC

Visual Understanding

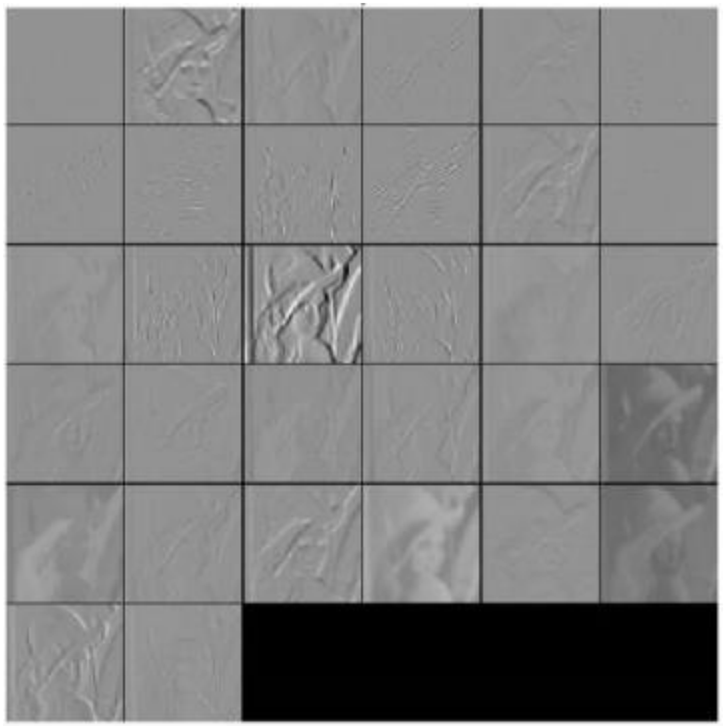


Visual Understanding

- Conv layer 1



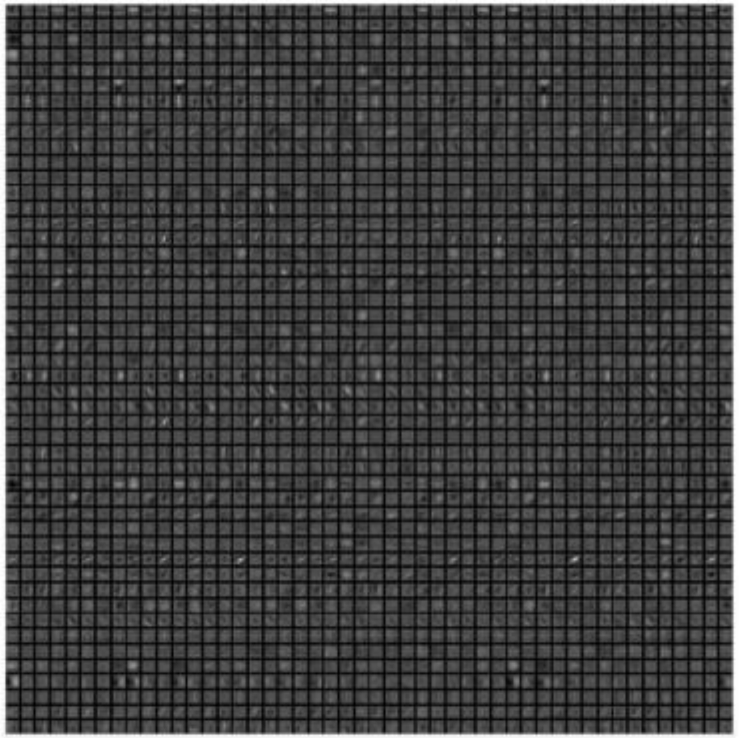
filters



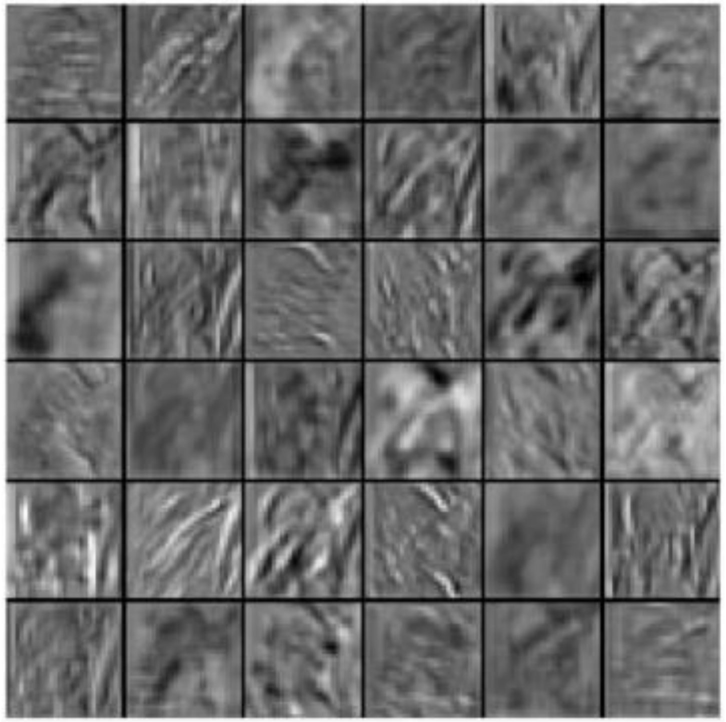
data

Visual Understanding

- Conv layer 2

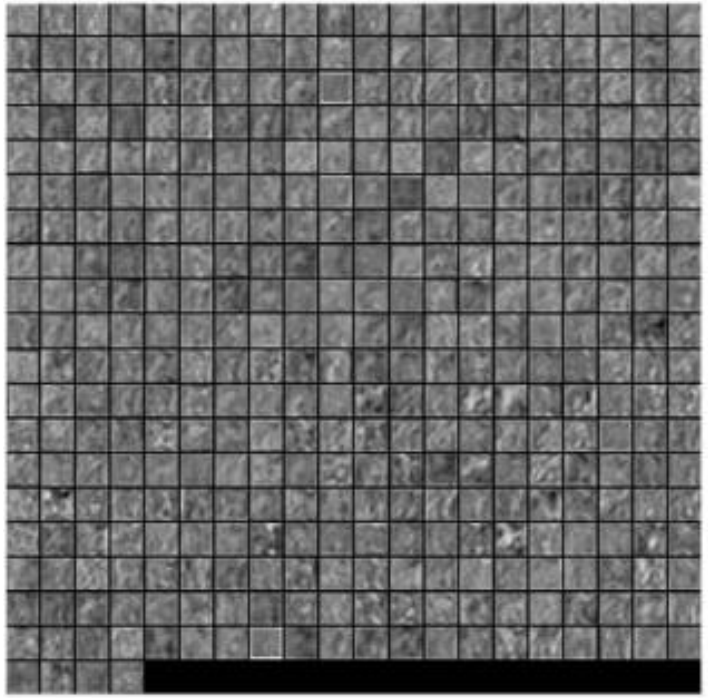


filters

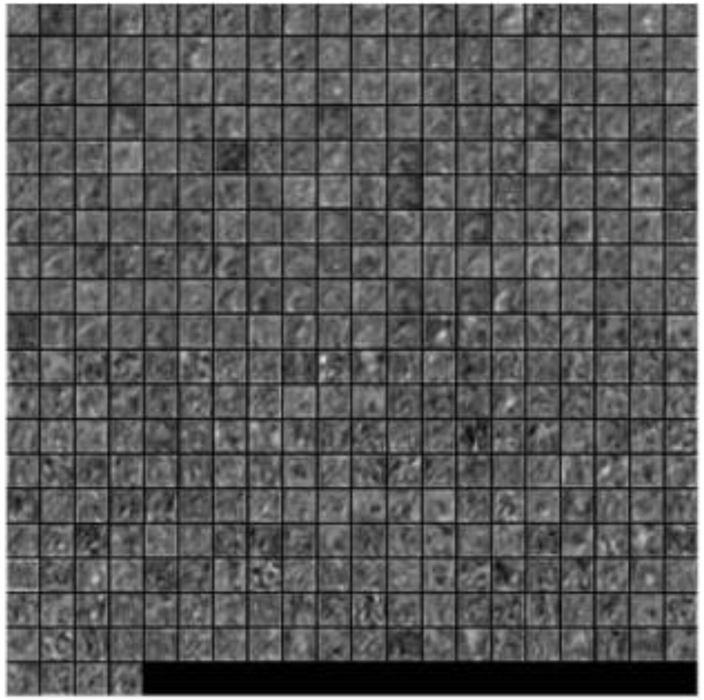


data

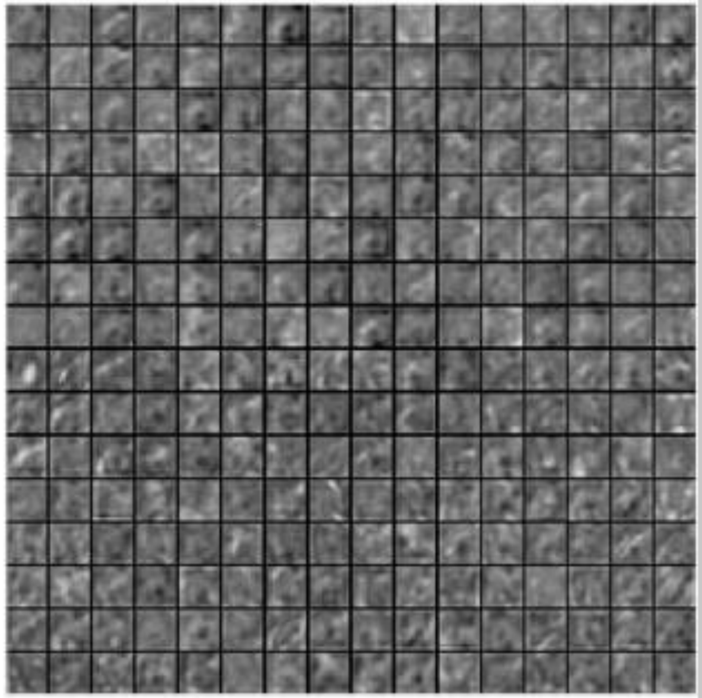
Visual Understanding



Conv3 layer data



Conv4 layer data



Conv5 layer data

Training Algorithm

$$\frac{dy}{dt} = \frac{dy}{dx}\frac{dx}{dt}$$

- Similar to the machine learning algorithm:
  - define "Loss funcion", measure the different from real results
- Find minimum loss function W and B, CNN use the SGD
- SGD need to compute the partial derivative(偏导) of W and B, iterate and update W and B.
- BP algorithm used to compute the partial derivative
- BP algorithm core is chain rule to compute the different level dW and db.

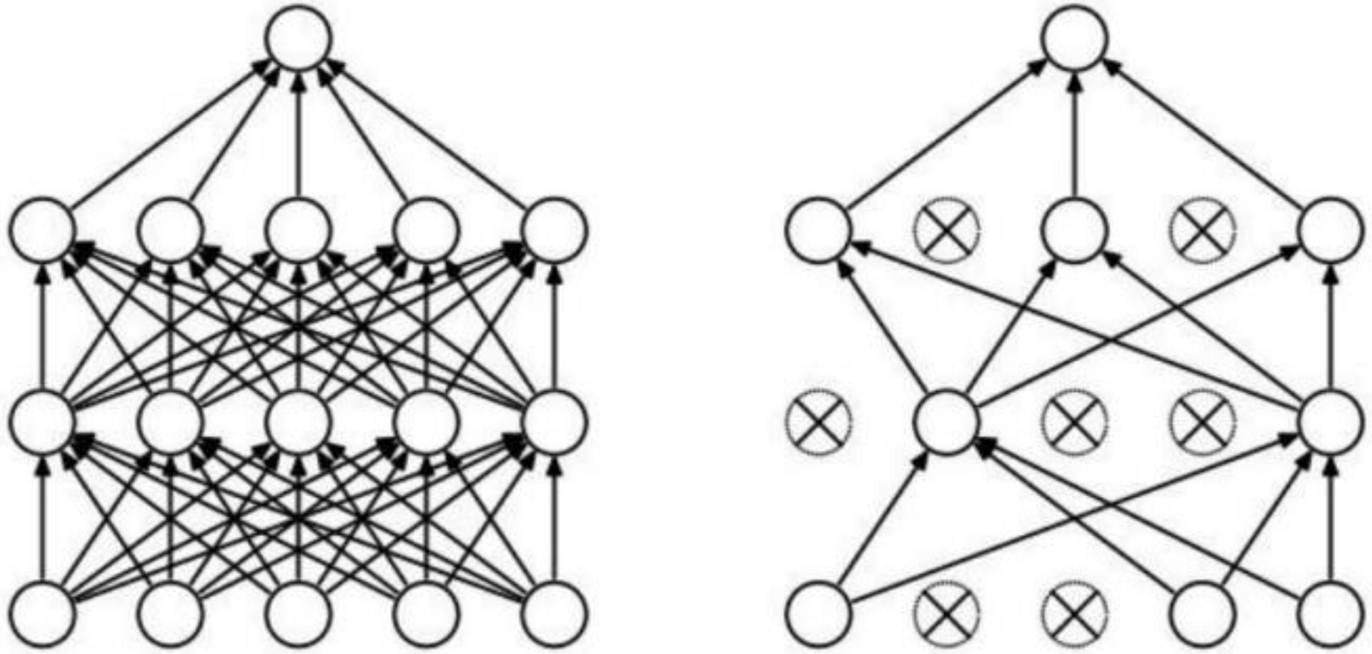
$$\frac{\partial y}{\partial x_i} = \sum_{t=1}^{m} \frac{\partial y}{\partial u_l}\frac{\partial u_l}{\partial x_i}$$

Advantages and Disadvantages

- Advantages:
  1. Shared convolution kernel, easy to handle high dimensional data
  2. Need not to manual extract feature.
  3. Train the weights ===>Get feature
  4. Deep network gets more images informations.
- Disadvantages:
  1. Need to adjust the parameters, need many sample data, need GPU
  2. Physical meaning is not clear

Regularization and Dropout

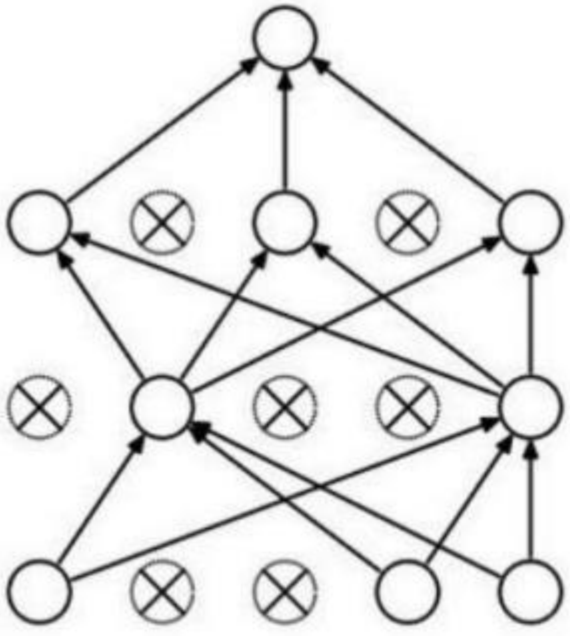
- ➢Neural network learning ability is strong and maybe leads to overfitting.
- ➢Regularization:Dropout
- “randomly set some neurons to zero in the forward pass”

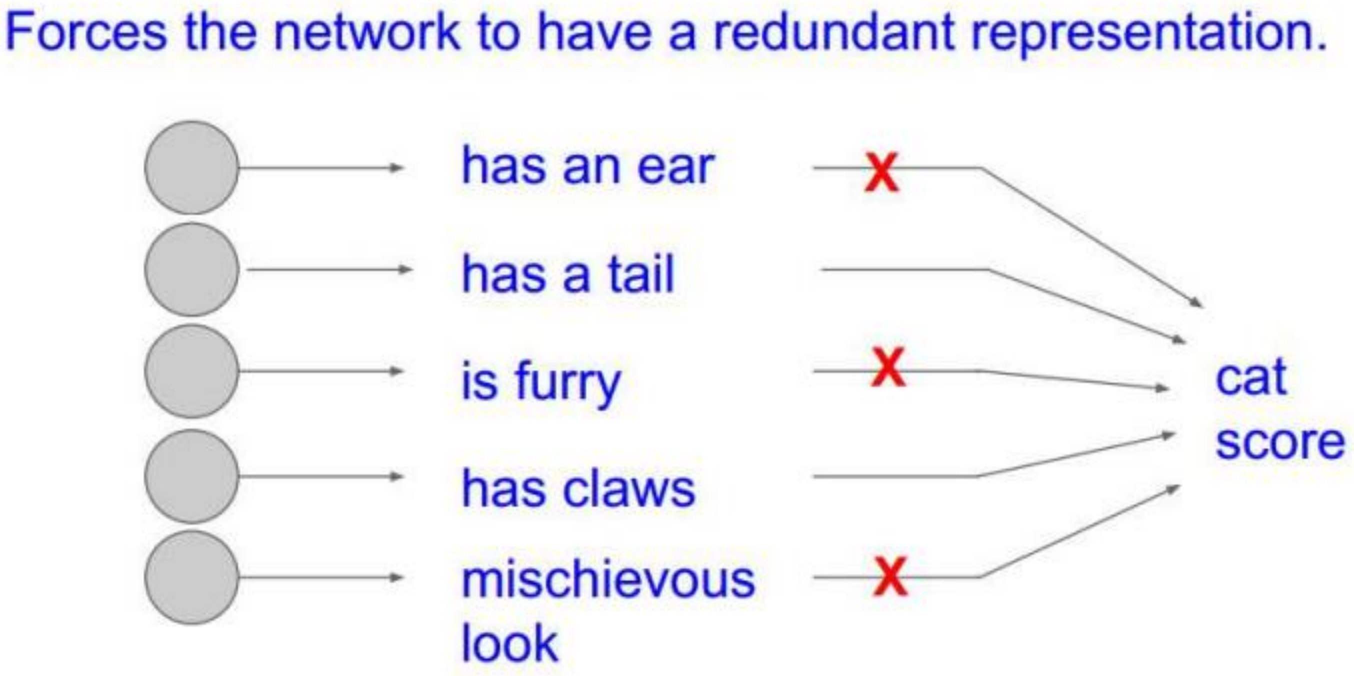


(a) Standard Neural Net　　(b) After applying dropout.

Regularization and dropout

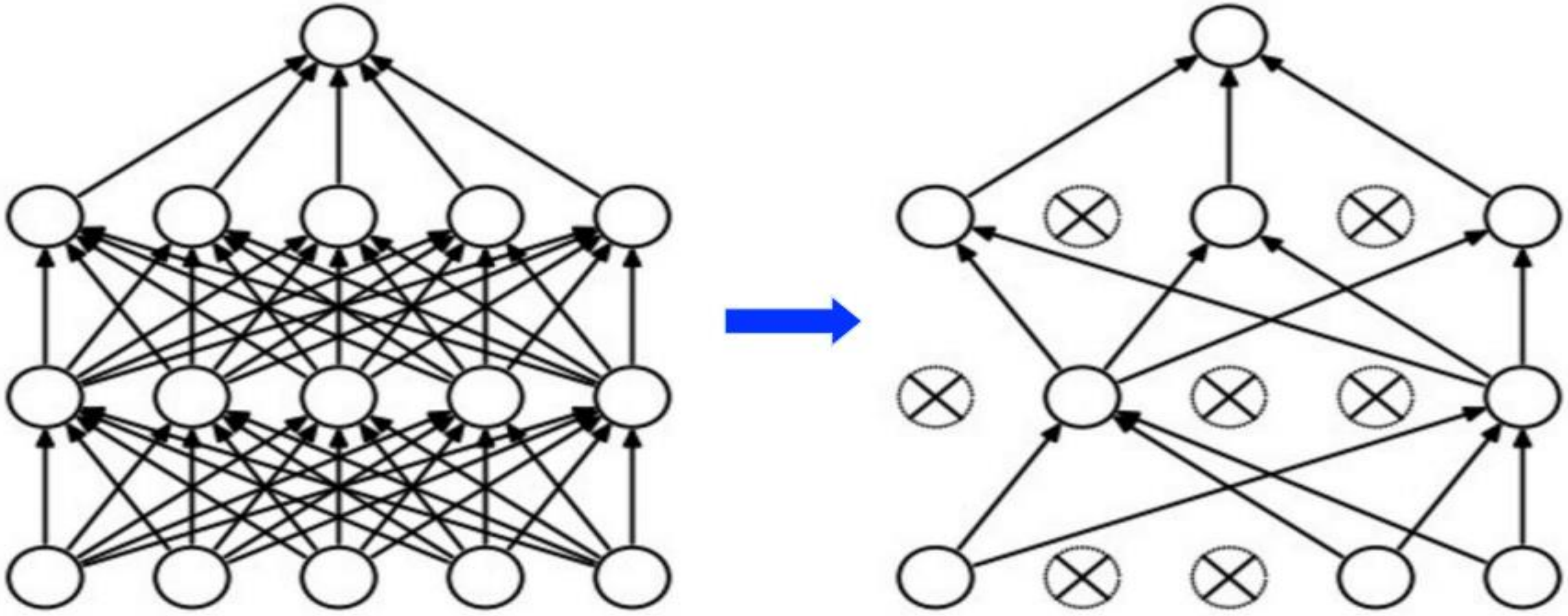
- How to understand prevent overfitting?
  - Don't let the neural network to remember too much information
  - The only cat, need generalization ability





Regularization and Dropout

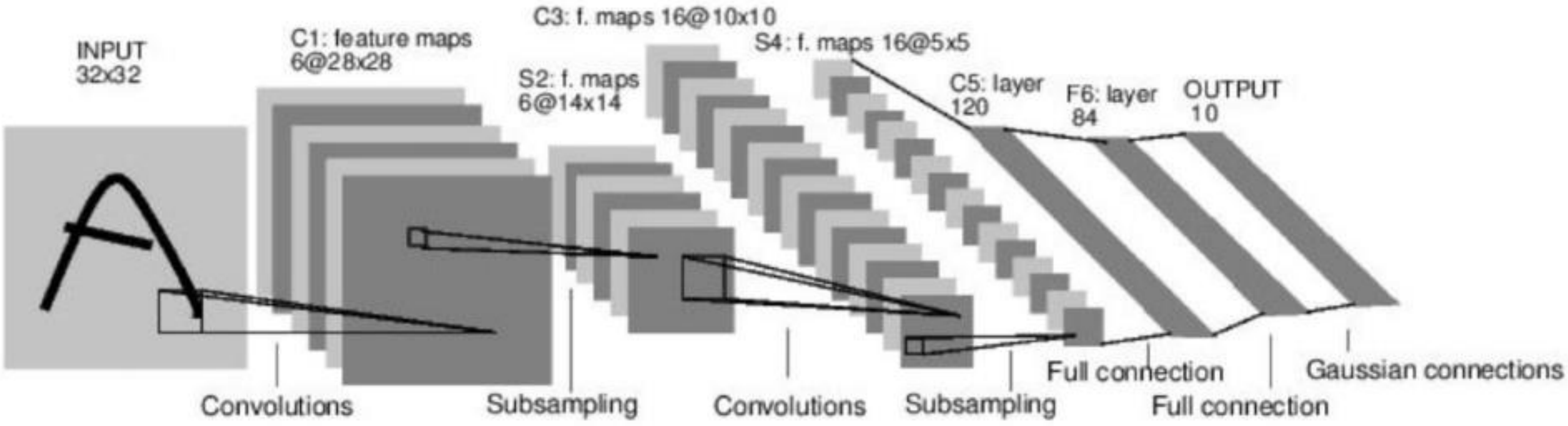
➢ How to understand prevent overfitting?
- Every time to close some perceptron(感知器), get the new model to make fuse(融合).
- Don't listen to the only expert's view.



Typical structure in Convolution Neural network

- LeNet, The first used for digital recognition
- AlexNet, Get the second in 2012 ILSVRC, deeper than the LeNet, use the Multi-level small convolution layers to replace the total convolution layers.
- ZF Net, The champion in 2013 ILSVRC
- GoogLeNet The champion in 2014 ILSVRC
- VGGNet, The model in 2014 ILSVRC, not good than GoogleNet in image recognition, but it is effective in object detection.
- ResNet, The champion in 2015 ILSVRC, Structure modification (Residual learning) to adapt the deeper CNN training.

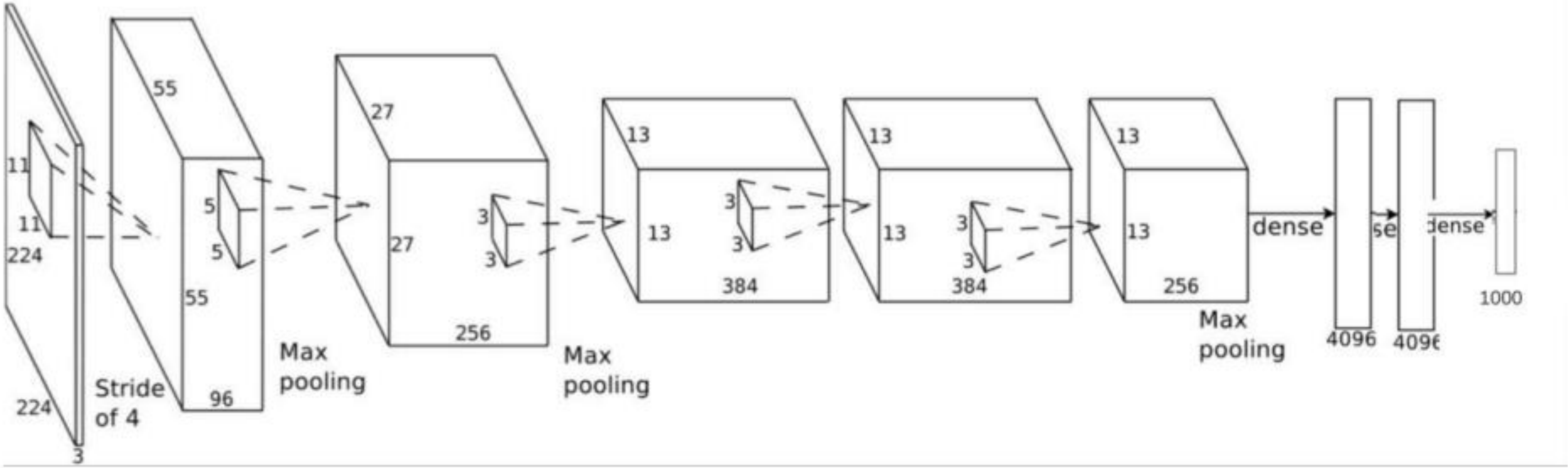
LeNet



Conv filters were 5x5, applied at stride 1
Subsampling (Pooling) layers were 2x2 applied at stride 2
i.e. architecture is [CONV-POOL-CONV-POOL-CONV-FC]

AlexNet

- The number of neurons in each layer is given by 253440, 186624, 64896, 64896, 43264, 4096, 4096, 1000

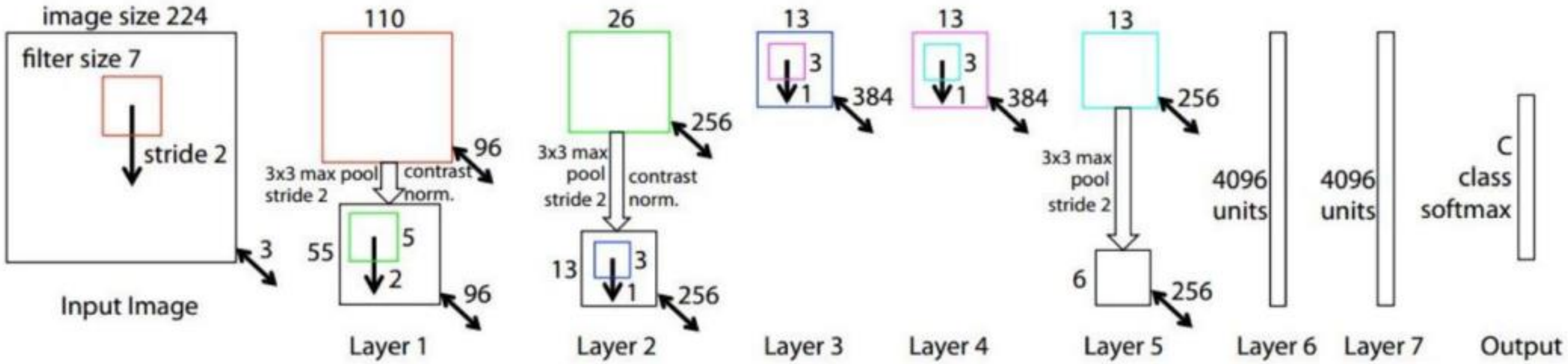


AlexNet

Full (simplified) AlexNet architecture:
[227x227x3] INPUT
[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization layer
[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization layer
[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)

Details/Retrospectives:
- first use of ReLU
- used Norm layers (not common anymore)
- heavy data augmentation
- dropout 0.5
- batch size 128
- SGD Momentum 0.9
- Learning rate 1e-2, reduced by 10 manually when val accuracy plateaus
- L2 weight decay 5e-4
- 7 CNN ensemble: 18.2% -> 15.4%

ZFNet



AlexNet but:
CONV1: change from (11x11 stride 4) to (7x7 stride 2)
CONV3,4,5: instead of 384, 384, 256 filters use 512, 1024, 512

ImageNet top 5 error: 15.4% -> 14.8%

VGGNet

```
INPUT: [224x224x3]
CONV3-64: [224x224x64]
CONV3-64: [224x224x64]
POOL2: [112x112x64]
CONV3-128: [112x112x128]
CONV3-128: [112x112x128]
POOL2: [56x56x128]
CONV3-256: [56x56x256]
CONV3-256: [56x56x256]
CONV3-256: [56x56x256]
POOL2: [28x28x256]
CONV3-512: [28x28x512]
CONV3-512: [28x28x512]
CONV3-512: [28x28x512]
POOL2: [14x14x512]
CONV3-512: [14x14x512]
CONV3-512: [14x14x512]
CONV3-512: [14x14x512]
POOL2: [7x7x512]
FC: [1x1x4096]
FC: [1x1x4096]
FC: [1x1x1000]
```

VGGNet

Only 3x3 CONV stride 1, pad 1
and 2x2 MAX POOL stride 2

best model

11.2% top 5 error in ILSVRC 2013
->
7.3% top 5 error

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

VGGNet

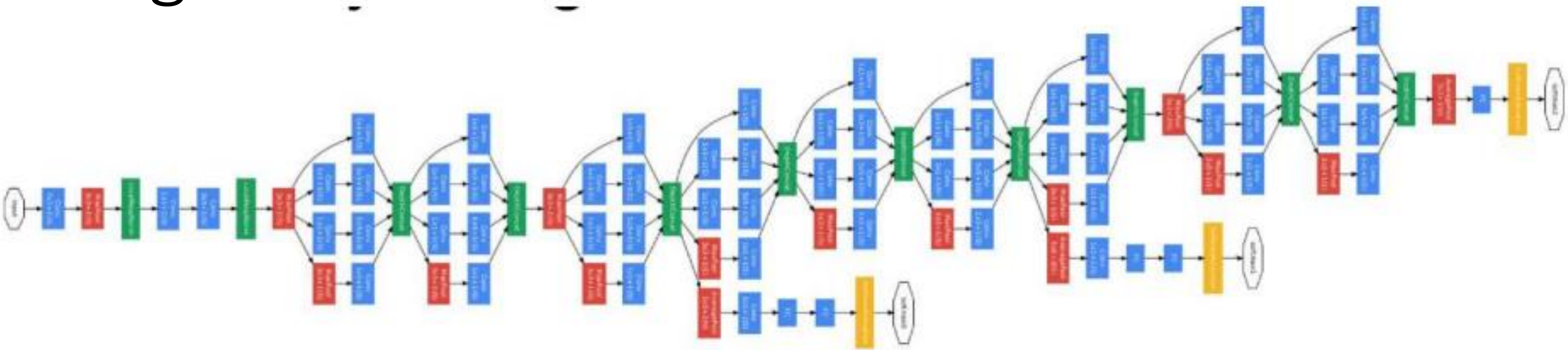
(not counting biases)

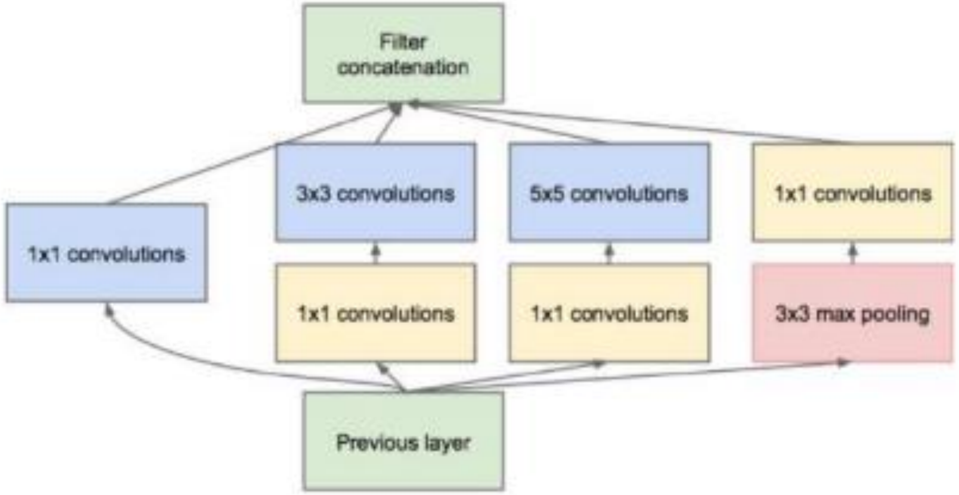
INPUT: [224x224x3] memory: 224*224*3=150K params: 0
CONV3-64: [224x224x64] memory: 224*224*64=3.2M params: (3*3*3)*64 = 1,728
CONV3-64: [224x224x64] memory: 224*224*64=3.2M params: (3*3*64)*64 = 36,864
POOL2: [112x112x64] memory: 112*112*64=800K params: 0
CONV3-128: [112x112x128] memory: 112*112*128=1.6M params: (3*3*64)*128 = 73,728
CONV3-128: [112x112x128] memory: 112*112*128=1.6M params: (3*3*128)*128 = 147,456
POOL2: [56x56x128] memory: 56*56*128=400K params: 0
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*128)*256 = 294,912
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*256)*256 = 589,824
CONV3-256: [56x56x256] memory: 56*56*256=800K params: (3*3*256)*256 = 589,824
POOL2: [28x28x256] memory: 28*28*256=200K params: 0
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*256)*512 = 1,179,648
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*512)*512 = 2,359,296
CONV3-512: [28x28x512] memory: 28*28*512=400K params: (3*3*512)*512 = 2,359,296
POOL2: [14x14x512] memory: 14*14*512=100K params: 0
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
CONV3-512: [14x14x512] memory: 14*14*512=100K params: (3*3*512)*512 = 2,359,296
POOL2: [7x7x512] memory: 7*7*512=25K params: 0
FC: [1x1x4096] memory: 4096 params: 7*7*512*4096 = 102,760,448
FC: [1x1x4096] memory: 4096 params: 4096*4096 = 16,777,216
FC: [1x1x1000] memory: 1000 params: 4096*1000 = 4,096,000

TOTAL memory: 24M * 4 bytes ~= 93MB / image (only forward! ~*2 for bwd)
TOTAL params: 138M parameters

ConvNet Configuration			
B	C	D	
13 weight layers	16 weight layers	16 weight layers	19
put (224 × 224 RGB image)			
conv3-64	conv3-64	conv3-64	cc
conv3-64	conv3-64	conv3-64	cc
maxpool			
conv3-128	conv3-128	conv3-128	co
conv3-128	conv3-128	conv3-128	co
maxpool			
conv3-256	conv3-256	conv3-256	co
conv3-256	conv3-256	conv3-256	co
	conv1-256	**conv3-256**	co
			co
maxpool			
conv3-512	conv3-512	conv3-512	co
conv3-512	conv3-512	conv3-512	co
	conv1-512	**conv3-512**	co
			co
maxpool			
conv3-512	conv3-512	conv3-512	co
conv3-512	conv3-512	conv3-512	co
	conv1-512	**conv3-512**	co
			co
maxpool			
FC-4096			
FC-4096			
FC-1000			
soft-max			

GoogLeNet





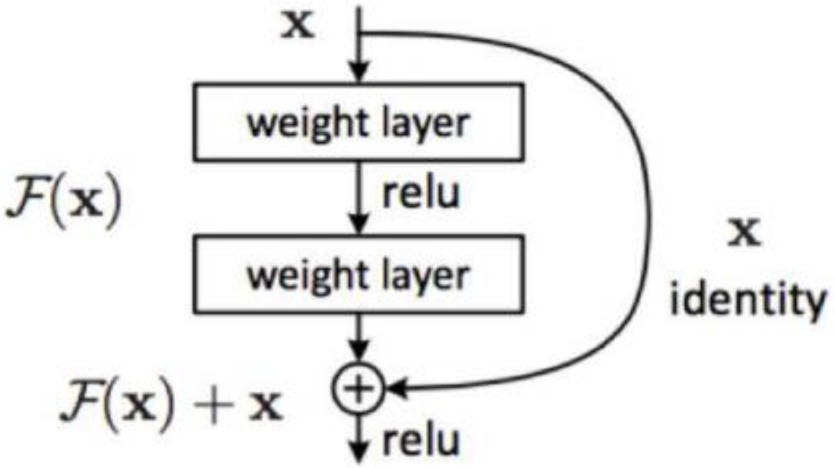
Inception module

ILSVRC 2014 winner (6.7% top 5 error)

ResNet

- ResNet, Deep Residual Learning network
- the champion in 2015 ILSVRC,Structure modification
- More than 8 times the length of the VGG

w



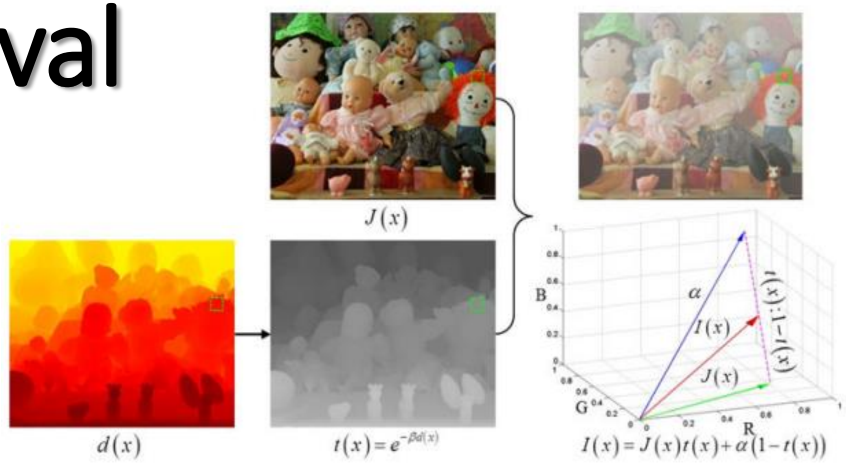
Dehaze Methods

- Dark Channel Prior
- Color Attenuation Prior
- DehazeNet
- MSCNN

DehazeNet:An End-to-End System for Single Image Haze Removal

- Bolun Cai, Xiangmin Xu, Member, IEEE, Kui Jia, Member, IEEE,
- Chunmei Qing, Member, IEEE, and Dacheng Tao, Fellow, IEEE

DehazeNet:An End-to-End System for Single Image Haze Removal



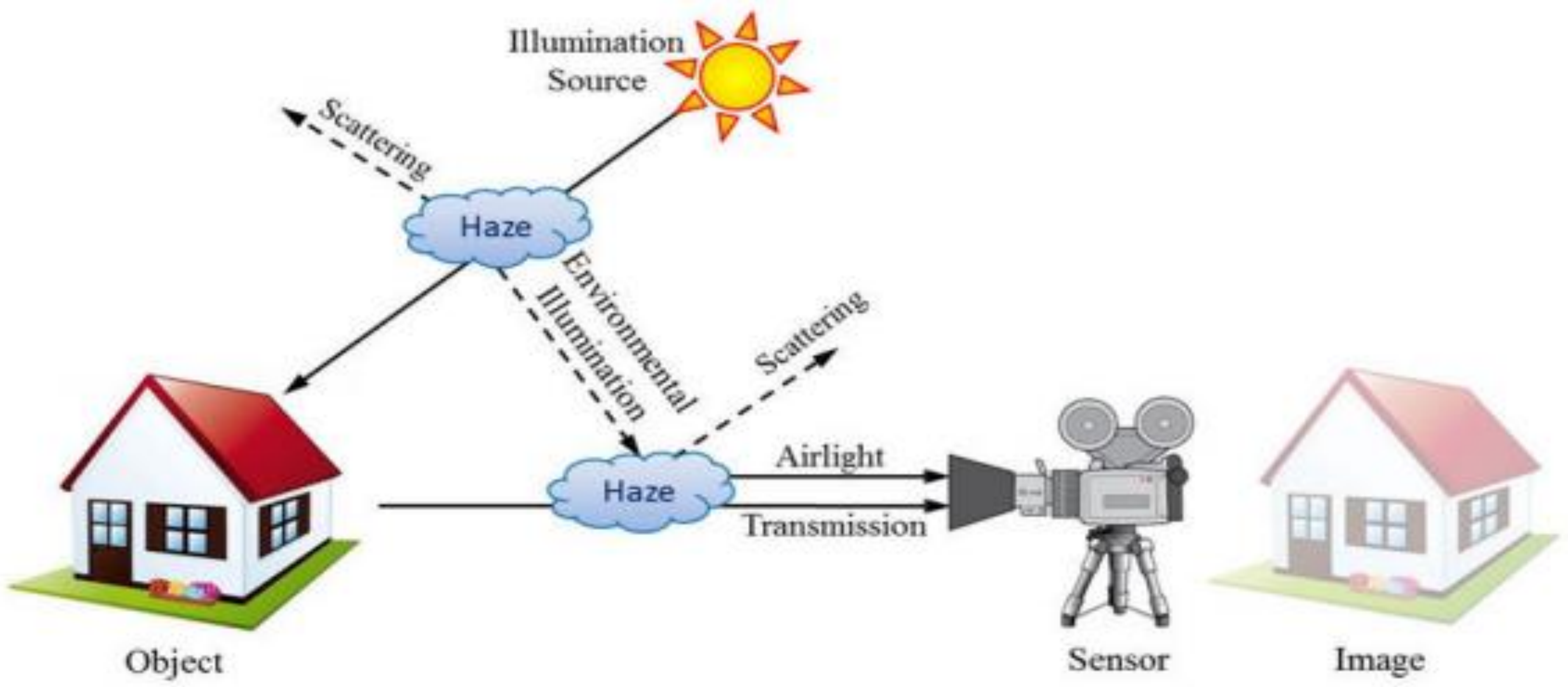
Estimation of a global atmospheric light

Recover an accurate medium transmission map

DehazeNet, a trainable CNN based end-to-end system for medium transmission estimation

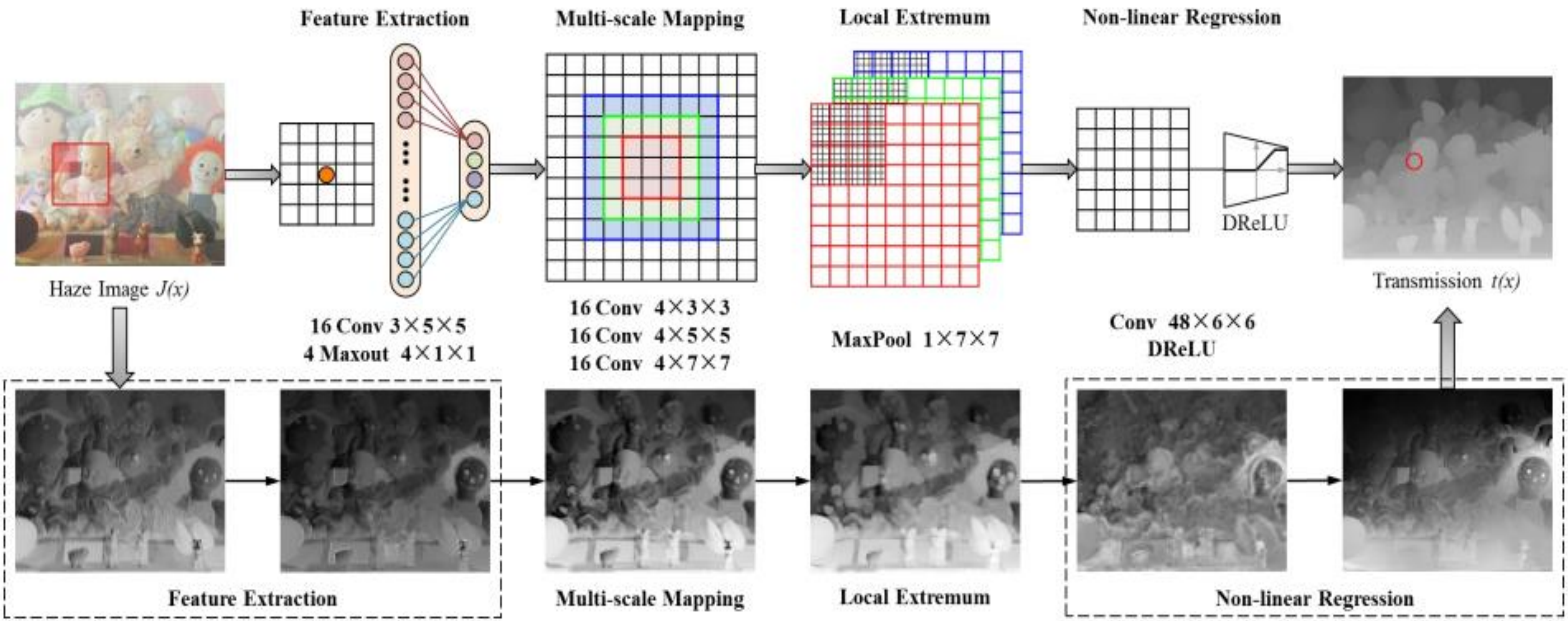
- Bolun Cai, Xiangmin Xu, Member, IEEE, Kui Jia, Member, IEEE,
- Chunmei Qing, Member, IEEE, and Dacheng Tao, Fellow, IEEE

DehazeNet:An End-to-End System for Single Image Haze Removal



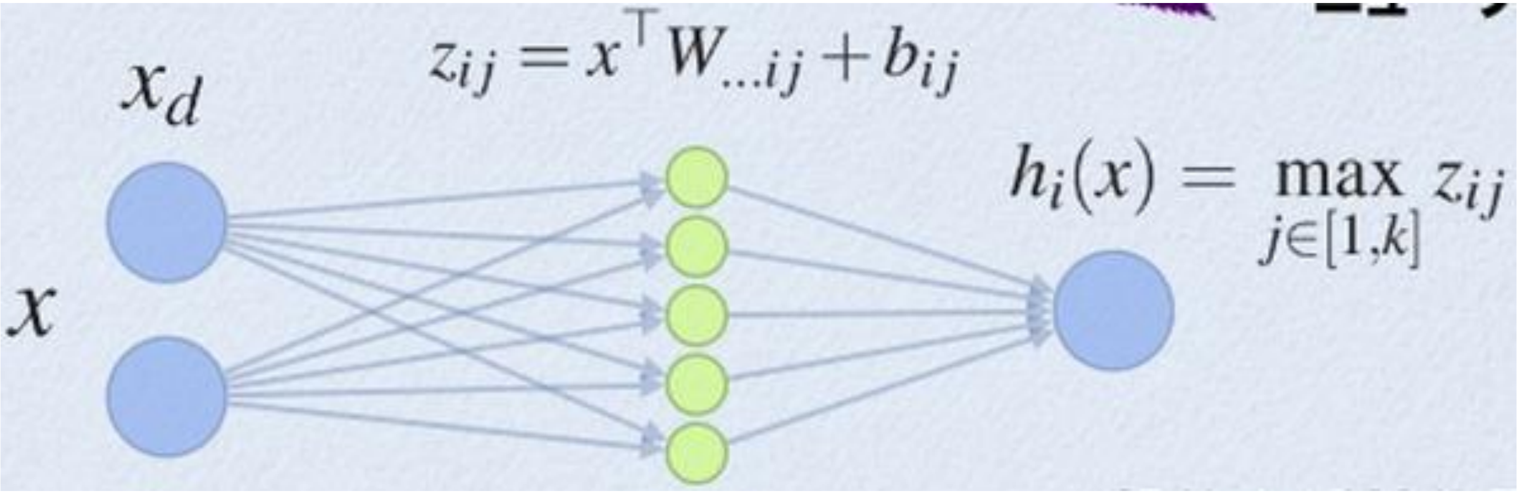
DehazeNet:An End-to-End System for Single Image Haze Removal

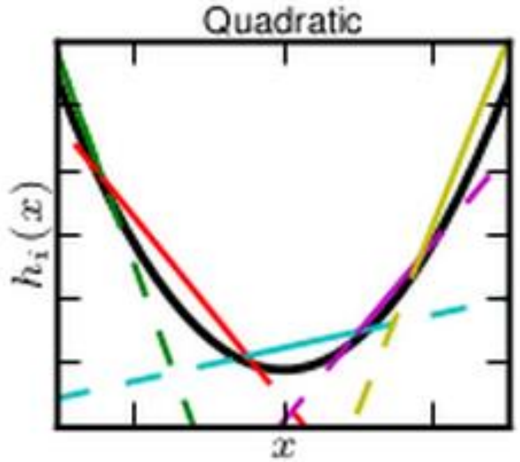
1. **Learn and estimate the mapping relations between hazy image patches and their medium transmissions.**
2. **Propose a novel nonlinear activation function in Net, called Bilateral Rectified Linear Unit(BReLU)**
3. **Establish connections between components of DehazeNet and priors used in existing dehazing methods.**



- **DehazeNet conceptually consists of four sequential operations (feature extraction, multi-scale mapping, local extremum and non-linear regression), which is constructed by 3 convolution layers, a max-pooling, a Maxout unit and a BReLU activation function.**

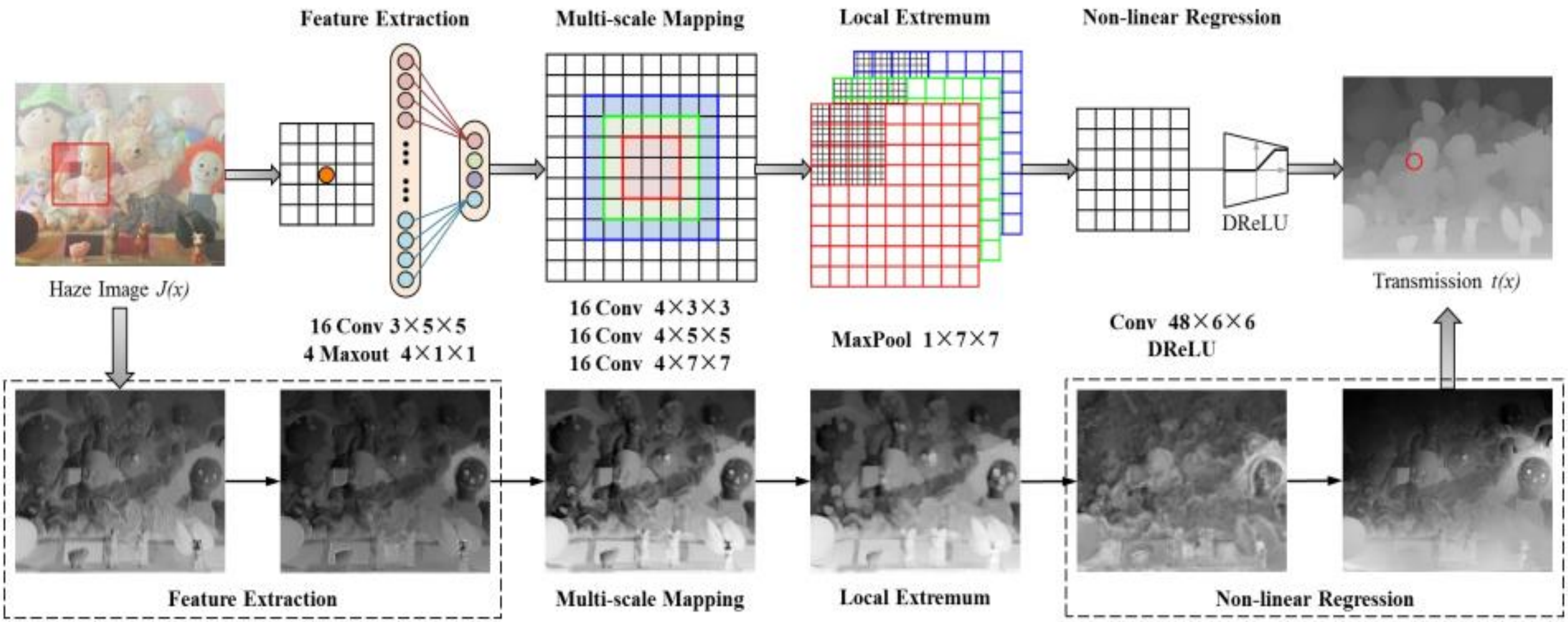
DehazeNet:An End-to-End System for Single Image Haze Removal





$$h_i(x) = \max_{j \in [1,k]} z_{ij} \qquad z_{ij} = x^T W_{\dots ij} + b_{ij}, and W \in R^{d*m*k}$$

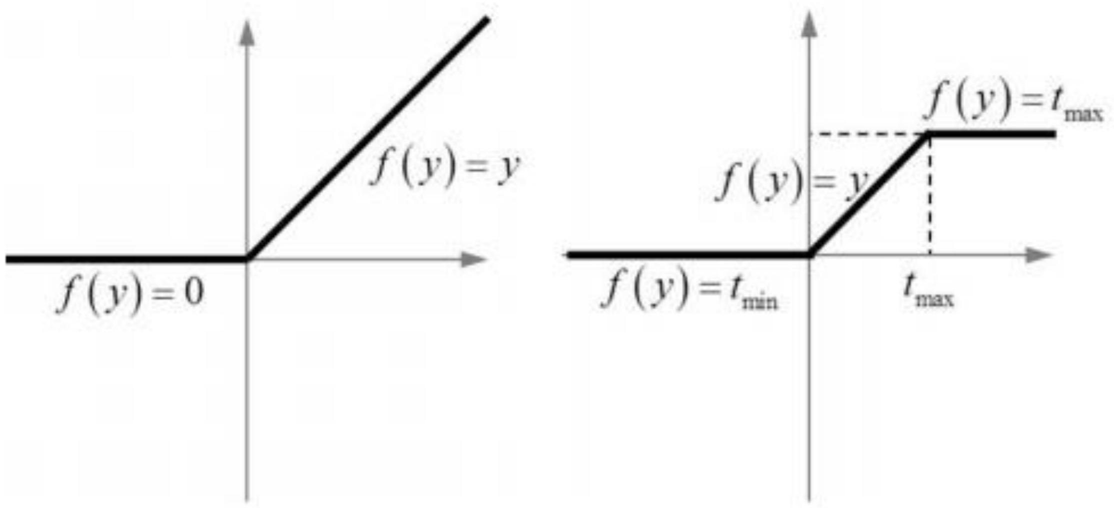
http://www.cnblogs.com/tornadomeet/p/3428843.html



- **DehazeNet conceptually consists of four sequential operations (feature extraction, multi-scale mapping, local extremum and non-linear regression), which is constructed by 3 convolution layers, a max-pooling, a Maxout unit and a BReLU activation function.**

DehazeNet:An End-to-End System for Single Image Haze Removal

- **Bilateral restraint:**
  - It applies a priori constraint (先验约束) to reduce the solution space scale.
- **Local linearity:**
  - It overcomes the gradient vanishing(梯度消失) to gain better precision.



(a) ReLU (b) BReLU

DehazeNet:An End-to-End System for Single Image Haze Removal

1. **Layer Designs of DehazeNet**
2. **Multi-scale Mapping**
3. **Local Extremum**
4. **Non-linear Regression**

Formulation	Type	Input Size	Num n	Filter $f \times f$	Pad
Feature Extraction	Conv	$3 \times 16 \times 16$	16	5×5	0
	Maxout	$16 \times 12 \times 12$	4	–	0
Multi-scale Mapping	Conv	$4 \times 12 \times 12$	16	3×3	1
			16	5×5	2
			16	7×7	3
Local Extremum	Maxpool	$48 \times 12 \times 12$	–	7×7	0
Non-linear Regression	Conv	$48 \times 6 \times 6$	1	6×6	0
	BReLU	1×1	1	–	0

DehazeNet:An End-to-End System for Single Image Haze Removal

Training of DehazeNet

1.Training data:

Two assumptions:

1)Image content is independent of medium transmission (the same image content can appear at any depths of scenes)

2)Medium transmission is locally constant (image pixels in a small patch tend to haze similar depths)

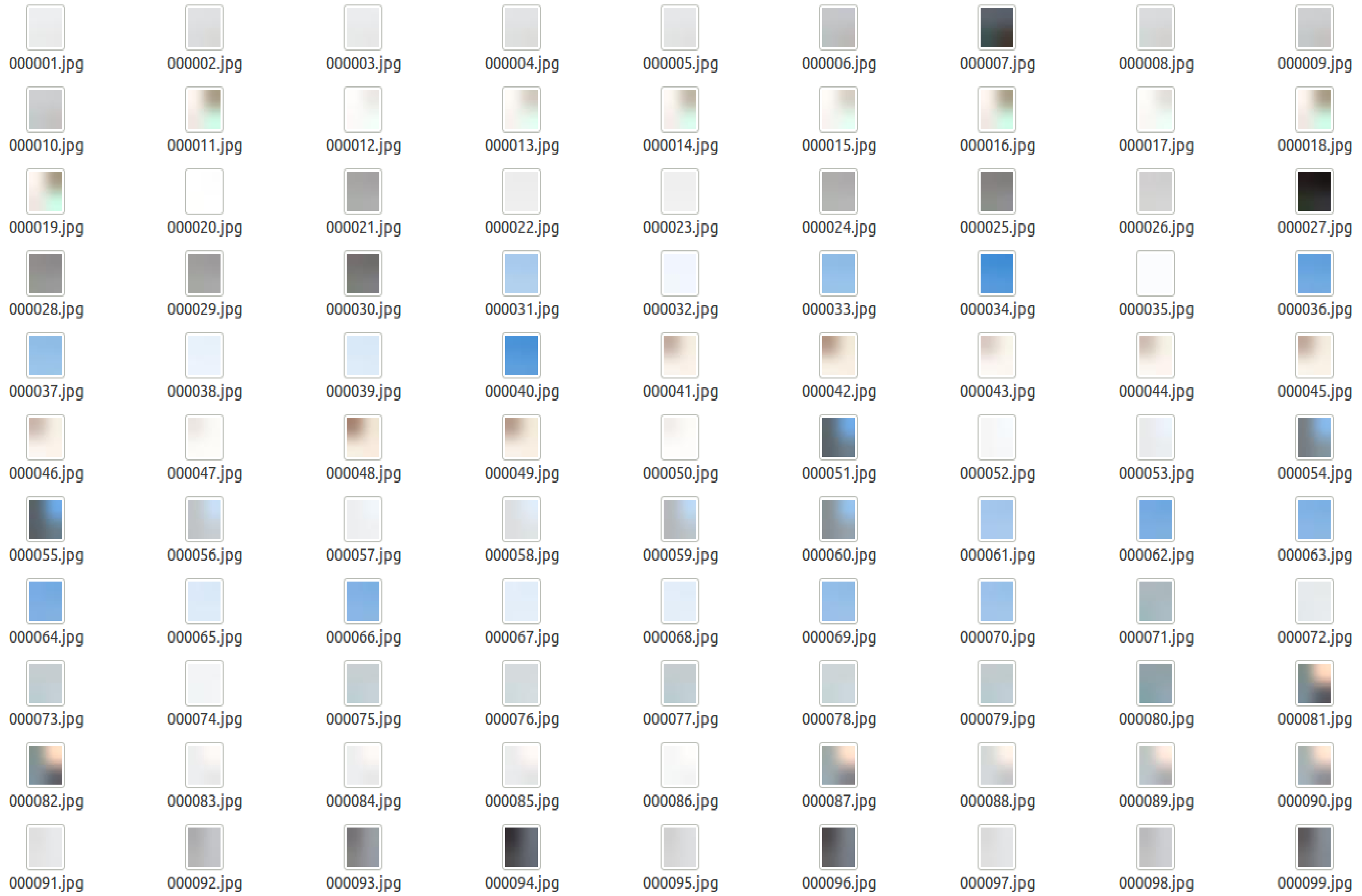
DehazeNet:An End-to-End System for Single Image Haze Removal

Training of DehazeNet

1.Training data:

Assume an arbitrary(任意的) transmission for an individual image patch.

Given a haze-free patch J(x), the atmospheric light A, and random transmission t, a haze image is synthesized.




```
000001.jpg 0.10516242
000002.jpg 0.18558914
000003.jpg 0.11729032
000004.jpg 0.16907009
000005.jpg 0.14768056
000006.jpg 0.34322159
000007.jpg 0.96592681
000008.jpg 0.22404758
000009.jpg 0.29576940
000010.jpg 0.29701062
000011.jpg 0.76015258
000012.jpg 0.17341965
000013.jpg 0.40635436
000014.jpg 0.56143965
000015.jpg 0.37423851
000016.jpg 0.74566579
000017.jpg 0.25431978
000018.jpg 0.76111873
000019.jpg 0.77886669
000020.jpg 0.01341821
000021.jpg 0.37994670
000022.jpg 0.07683825
000023.jpg 0.06725106
000024.jpg 0.34487915
000025.jpg 0.52858843
000026.jpg 0.19900572
000027.jpg 0.95942324
000028.jpg 0.48010922
000029.jpg 0.40867907
000030.jpg 0.60275455
000031.jpg 0.45066727
000032.jpg 0.07839893
000033.jpg 0.58494760
000034.jpg 0.98030893
000035.jpg 0.02719647
000036.jpg 0.81766764
000037.jpg 0.57564936
000038.jpg 0.11626336
000039.jpg 0.19522155
000040.jpg 0.93690418
000041.jpg 0.50169761
000042.jpg 0.63414945
000043.jpg 0.32533
```

DehazeNet:An End-to-End System for Single Image Haze Removal

Training of DehazeNet

2.Training method:

1) In the DehazeNet, supervised learning requires the mapping relationship F between RGB value and medium transmission.

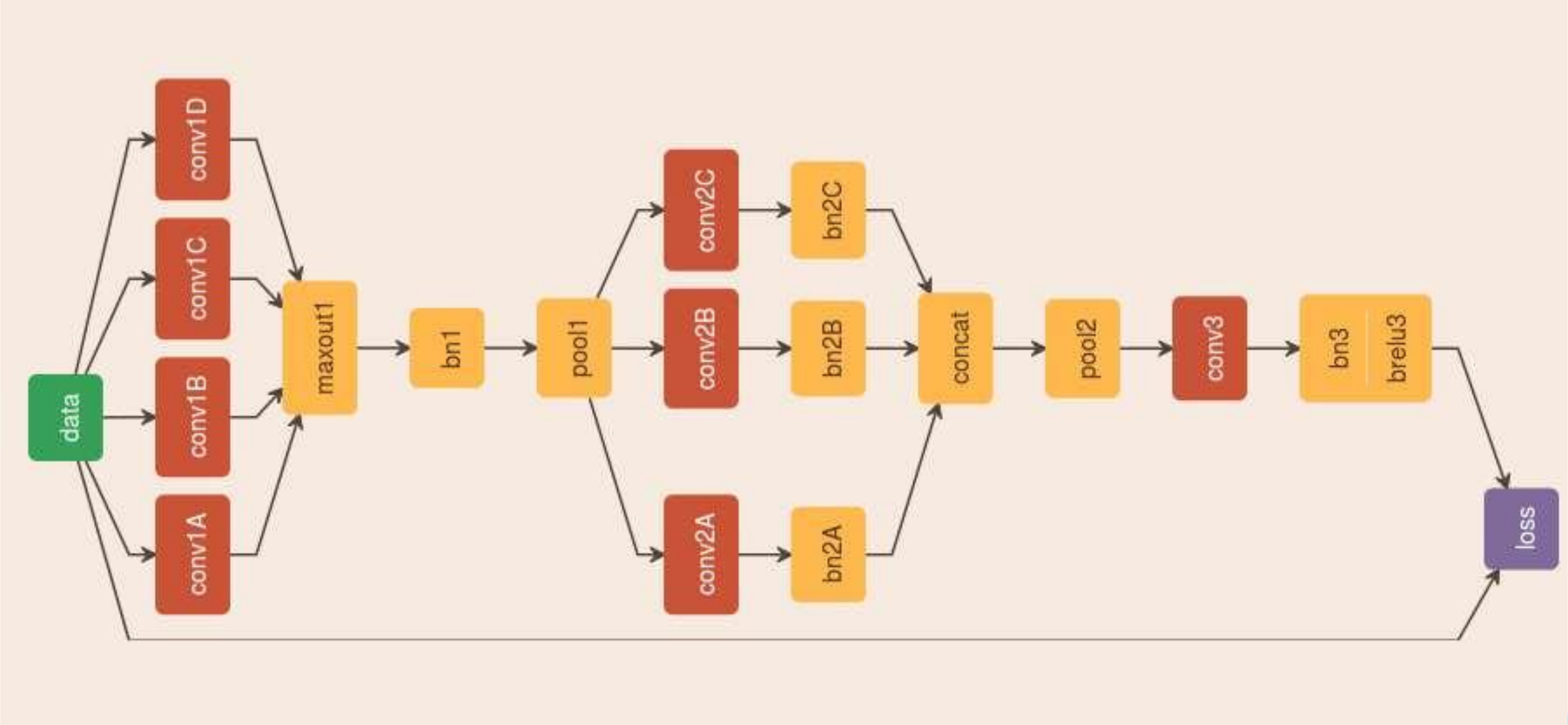
2) Network parameters θ = {W,B} are achieved through minimizing the loss function between the training patch I(x) and the corresponding ground truth medium transmission t.

3) MSE(Mean Squared Error) as the loss function:

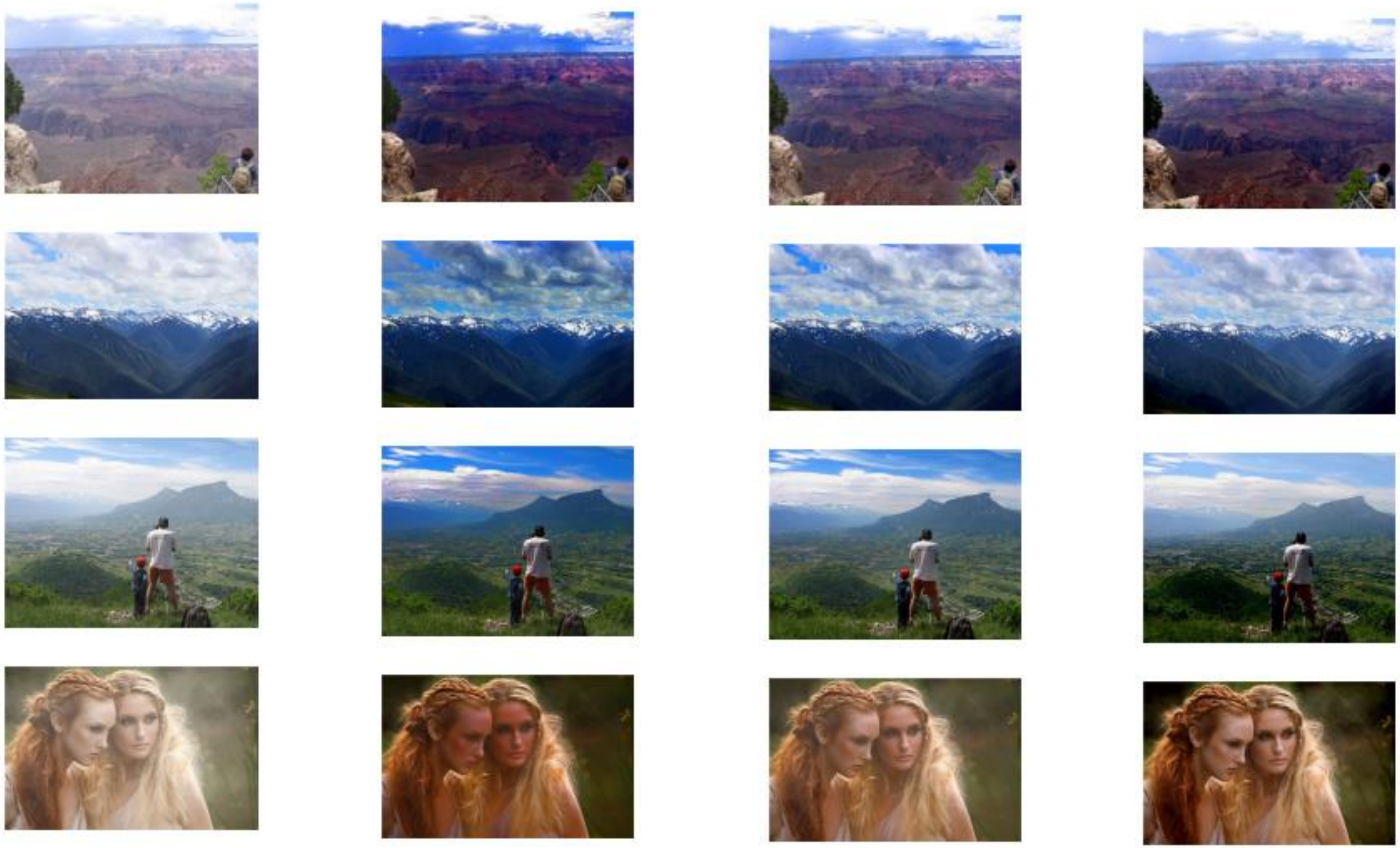
$$L(\theta)=\frac{1}{N}\sum_{i=1}^{N}\| F(I_i^P;\theta)-t_i \|^2$$

中国海洋大学

DehazeNet


data
conv1D
conv1C
conv1B
conv1A
maxout1
bn1
pool1
conv2C
conv2B
conv2A
bn2C
bn2B
bn2A
concat
pool2
conv3
bn3
brelu3
loss

Results



Hazy image　DCP　CAP　DehazeNet

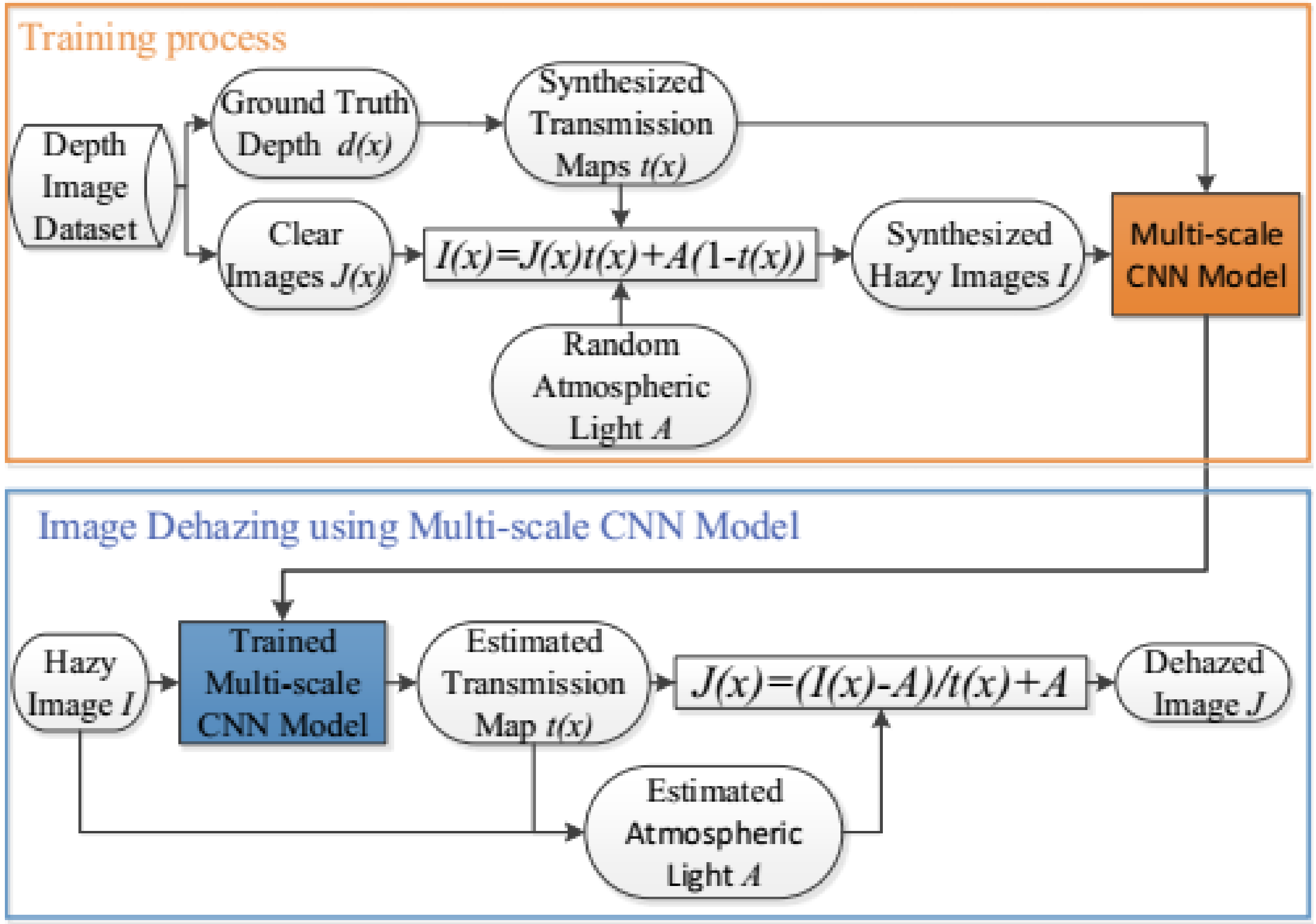
Dehaze Methods

- **Dark Channel Prior**
- **Color Attenuation Prior**
- **DehazeNet**
- **MSCNN**

Single Image Dehazing via Multi-Scale Convolutional Neural Network

- A multi-scale convolutional neural network for transmission estimation
- Analyze the differences between traditional hand-crafted features and the features learned by the CNN

Single Image Dehazing via Multi-Scale Convolutional Neural Network



(a)

- Train the multi-scale CNN based on synthesized dataset
- Predict transmission based on the trained network

Single Image Dehazing via Multi-Scale Convolutional Neural Network

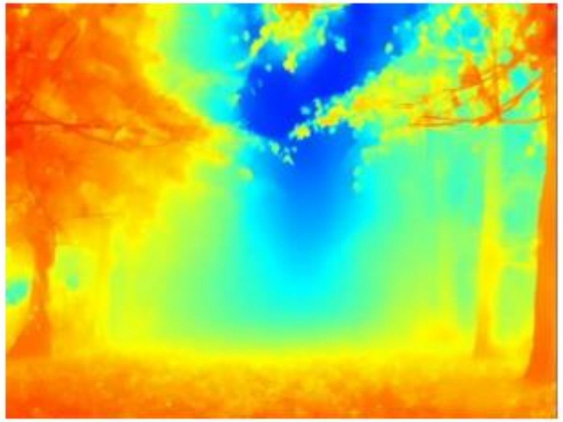
- Hazy imaging model

Atmospheric light

$$\boldsymbol{I}(x) = \boldsymbol{J}(x)t(x) + \boldsymbol{A}(1 - t(x))$$

Haze image

Transmission



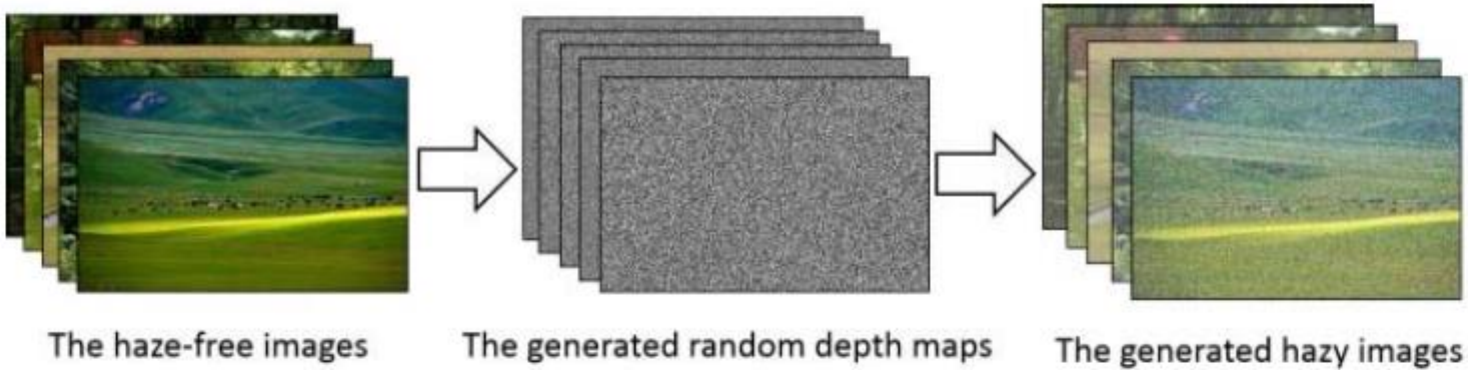
Scene

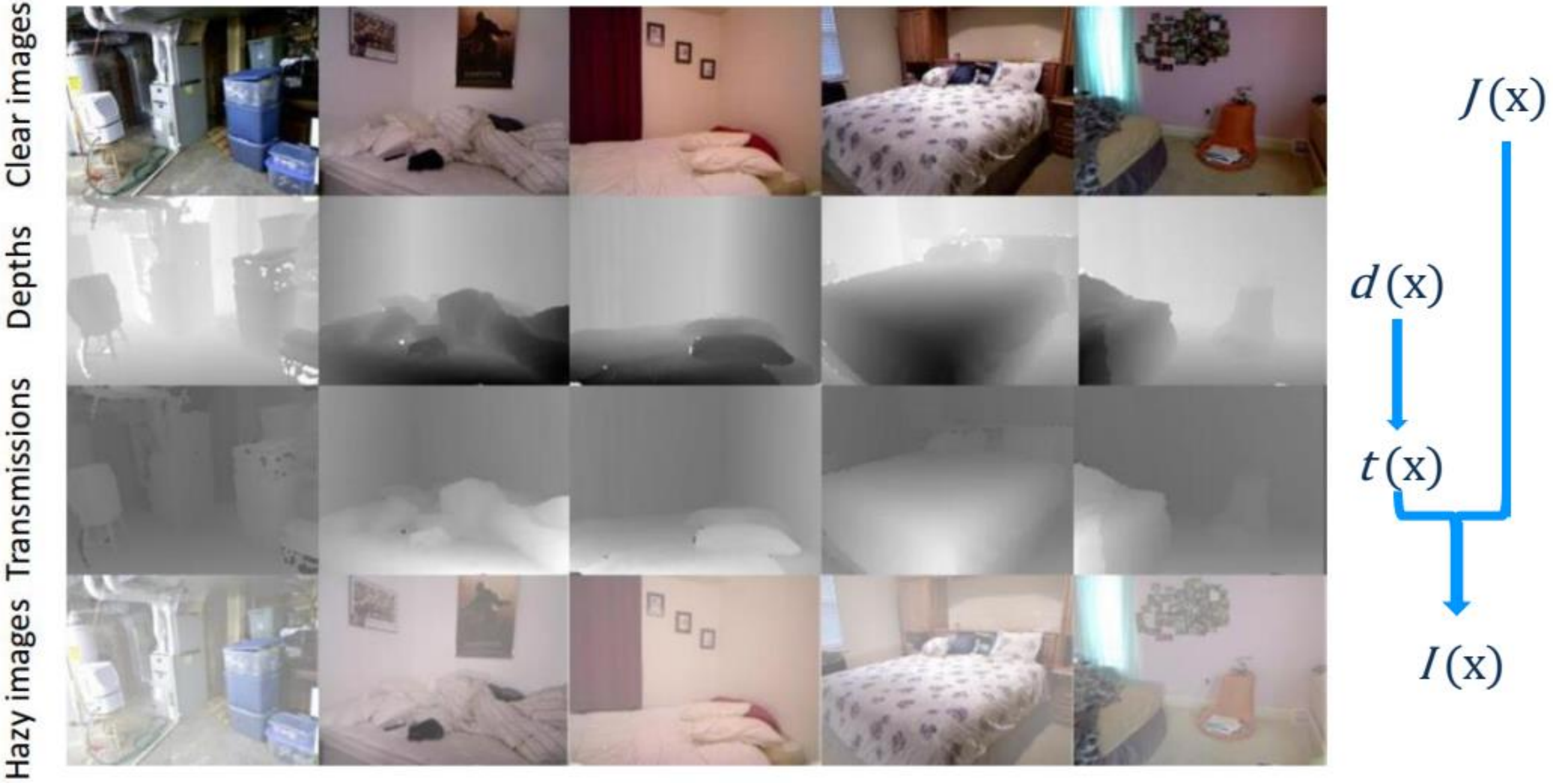
Single Image Dehazing via Multi-Scale Convolutional Neural Network

Scene depth recovery

1.Synthetic dataset

clear image + random depth maps + random global atmospheric light=sample haze images





Clear images
Depths
Transmissions
Hazy images
$J(x)$
$d(x)$
$t(x)$
$I(x)$

Single Image Dehazing via Multi-Scale Convolutional Neural Network

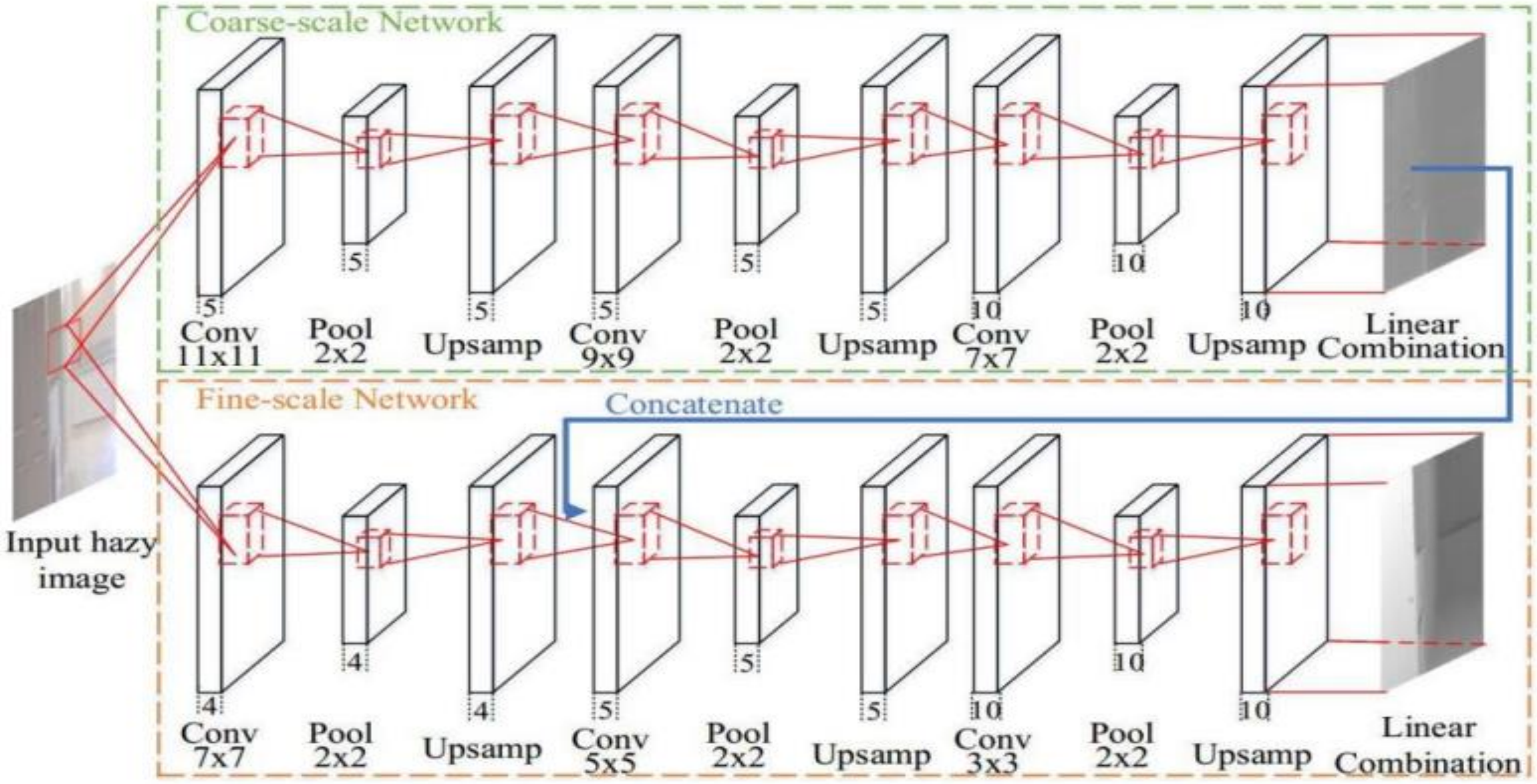
Clear image

Depth image

Hazy images with different β

Single Image Dehazing via Multi-Scale Convolutional Neural Network

- The scene transmission map is first estimated by a coarse-scale network and then refined be a fine-scale network.



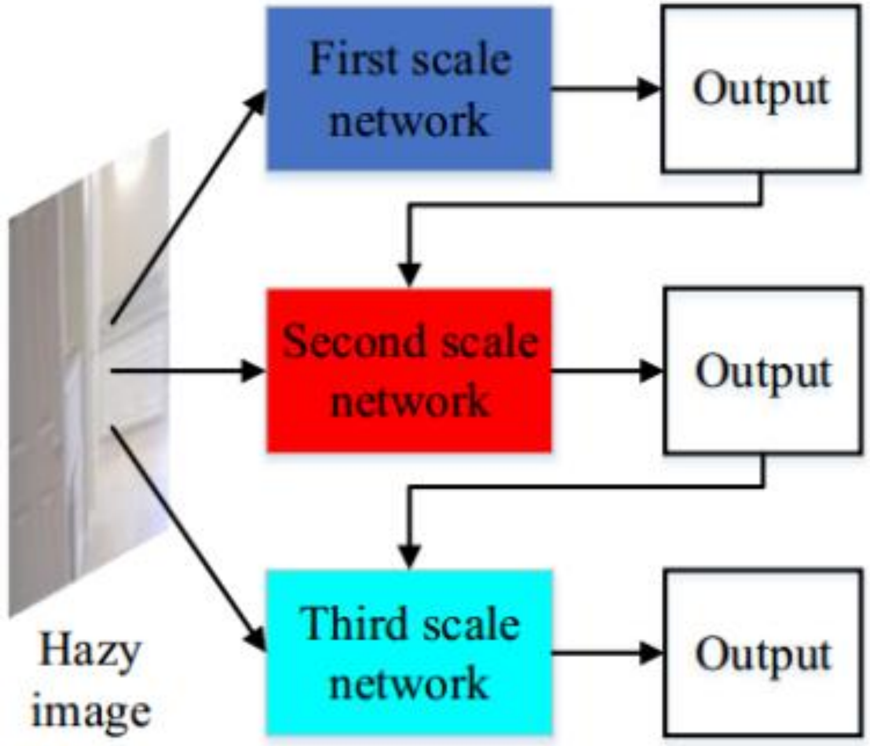
Single Image Dehazing via Multi-Scale Convolutional Neural Network

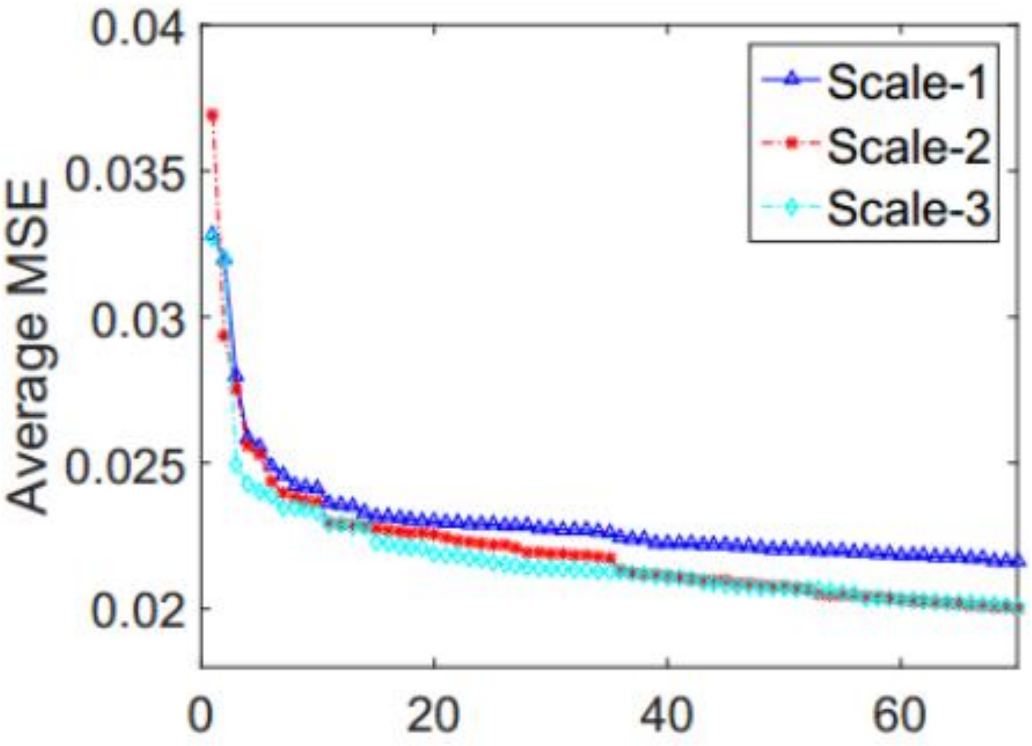
- Training loss

$$L(t_i(x), t_i^*(x)) = \frac{1}{q}\sum_{i=1}^{q} \| t_i(x) - t_i^*(x) \|^2$$

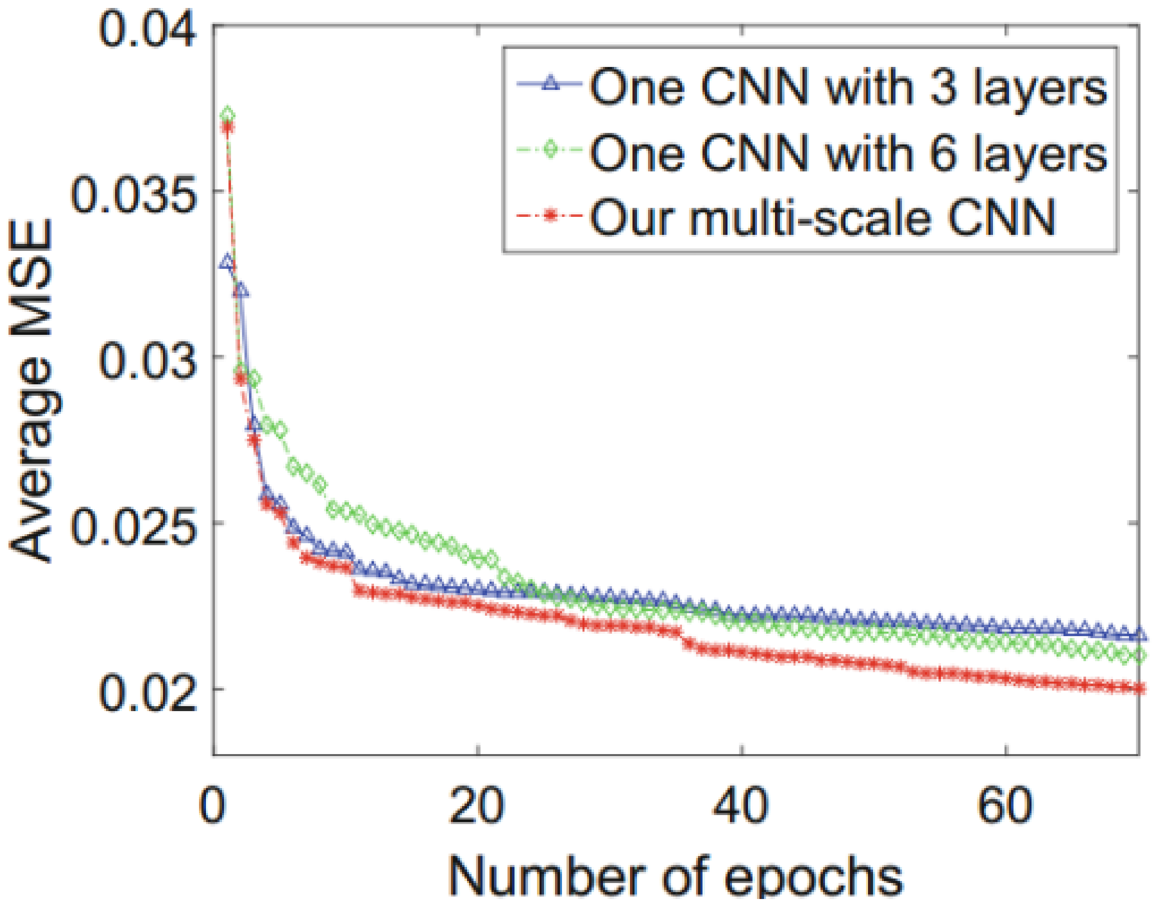
- The training loss is used in both coarse and fine-scale networks

Single Image Dehazing via Multi-Scale Convolutional Neural Network





Single Image Dehazing via Multi-Scale Convolutional Neural Network



Single Image Dehazing via Multi-Scale Convolutional Neural Network

- Compute atmospheric light from the estimated transmission map

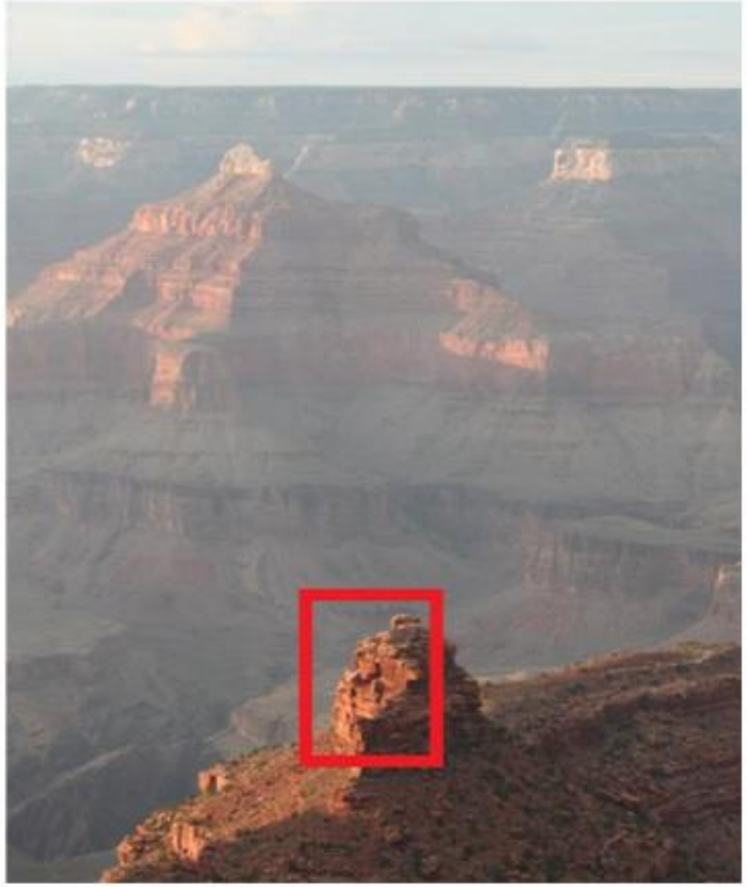
Single Image Dehazing via Multi-Scale Convolutional Neural Network

- Recover haze-free images after atmospheric light and transmission are estimated.

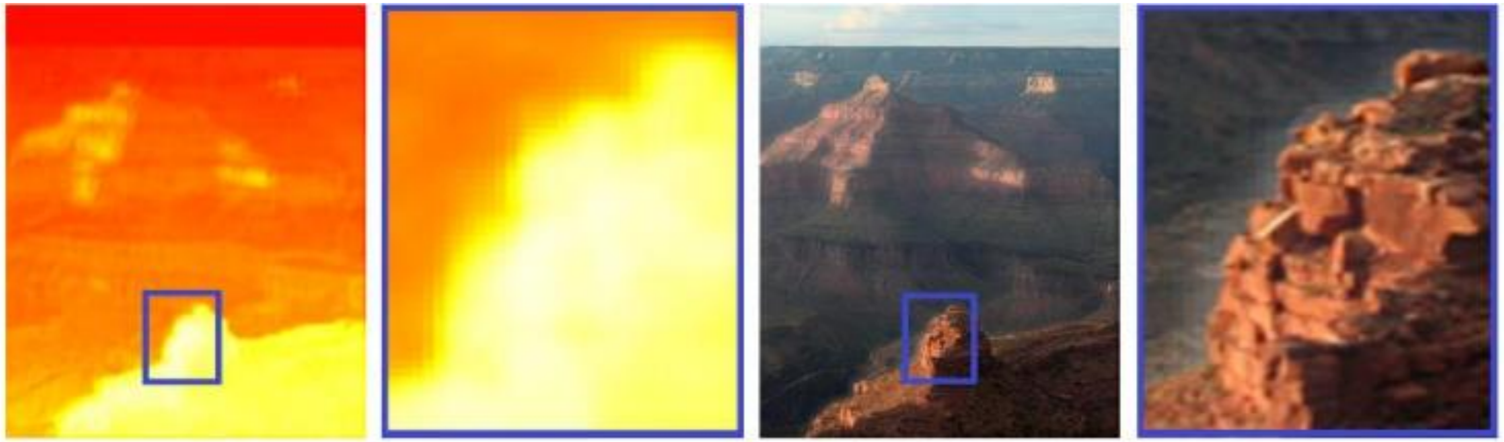
$$I(x) = J(x)t(x)+A(1\text{-}t(x))$$

$$J(x)=\frac{I(x)-A}{t(x)}+A \quad J\left(x\right)=\frac{I\left(x\right)-A}{\max\left(t\left(x\right),t_0\right)}+A$$

Single Image Dehazing via Multi-Scale Convolutional Neural Network



Hazy image



Without the fine-scale network

With the fine-scale network

Single Image Dehazing via Multi-Scale Convolutional Neural Network

Close shot Medium shot Long shot

Single Image Dehazing via Multi-Scale Convolutional Neural Network

- Failure case for nighttime haze images
- Nighttime hazy image model: $I(x) = J(x)t(x) + A(1 - t(x)) + L_a(x) * APSF$

Input

Proposed

Li et al.“Nighttime haze removal with glow and multiple light colors”. ICCV. 2015

Review

1 Review of deblurring method

2 Introduction of dehazing method

3 Introduction of Convolution Neural Network

4 Dehazing by Convolution Neural Network

THANKS