Interview Assignment

AI IMAGE DEGRADATION

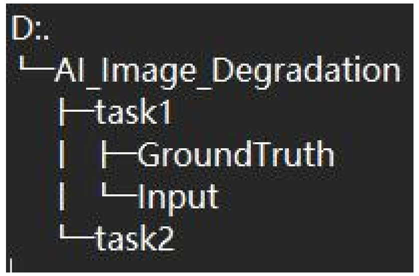
Yushuo Wang

Task 1

a. Algorithm

1. FineTune DeFlow[1] to generate paired data.
2. Use generated dataset to conduct super resolution task on ESRGAN[2].

Todo:
Dataset: 100 paired patches of size 512x512.
Use input folder data as distribution reference to generate degradation data (Can use DIV2K as clean data)
Use Ground Truth data to measure performance.


```
D:.
└─AI_Image_Degradation
    ├─task1
    |   ├─GroundTruth
    |   └─Input
    └─task2
```

[1] https://github.com/volflow/DeFlow
[2] https://github.com/xinntao/ESRGAN

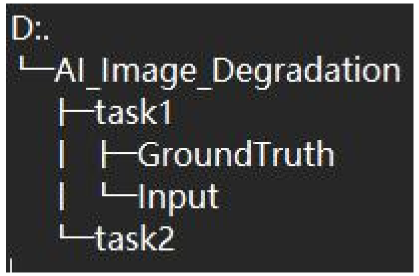
Task 2

a. Algorithm

1. FineTune DeFlow[1] to generate paired data.
2. Use generated dataset to conduct super resolution task on ESRGAN[2].

Todo:
Dataset: 120 frames of size 256X256 from real world data.
Solve the real world X2 super resolution problem.

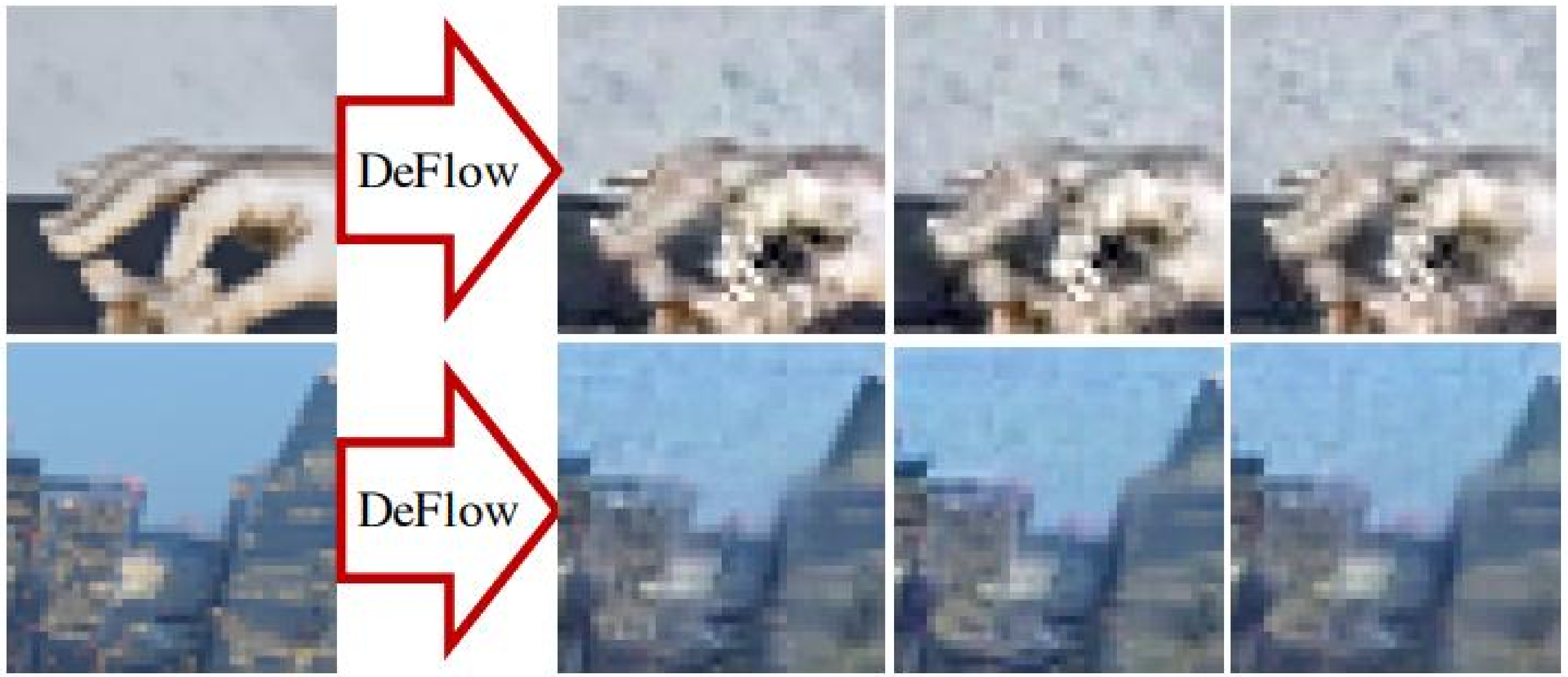
Design a new Domain Invariant Mapping Function h



```
D:.
└─AI_Image_Degradation
    ├─task1
    │   ├─GroundTruth
    │   └─Input
    └─task2
```

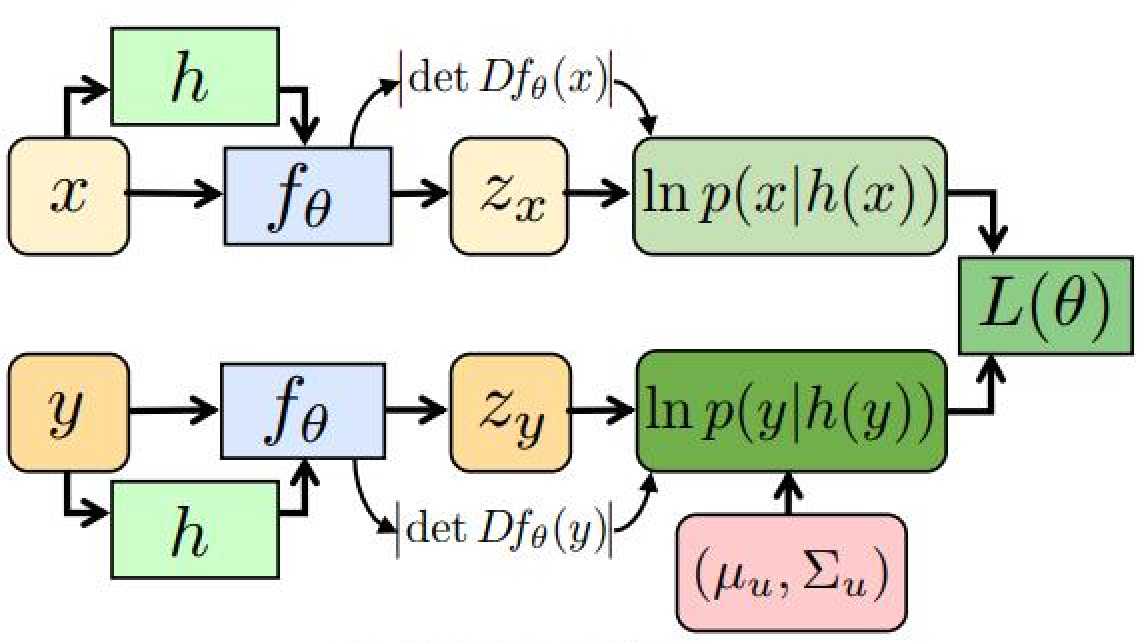
[1] https://github.com/volflow/DeFlow
[2] https://github.com/xinntao/ESRGAN

Model **DeFlow:** Learning Complex Image Degradations from Unpaired Data with Conditional Flows

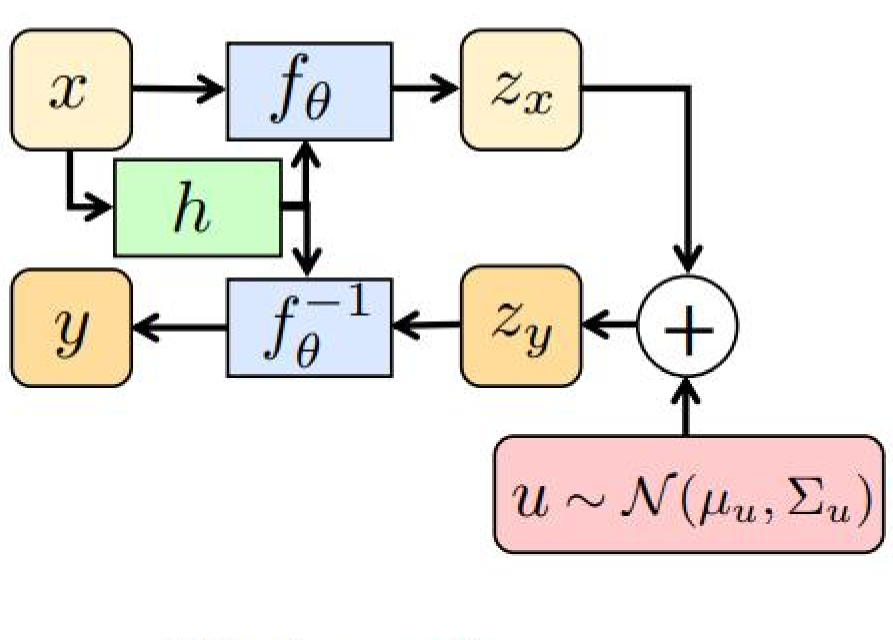


Model

DeFlow: Learning Complex Image Degradations from Unpaired Data with Conditional Flows



(a) Training

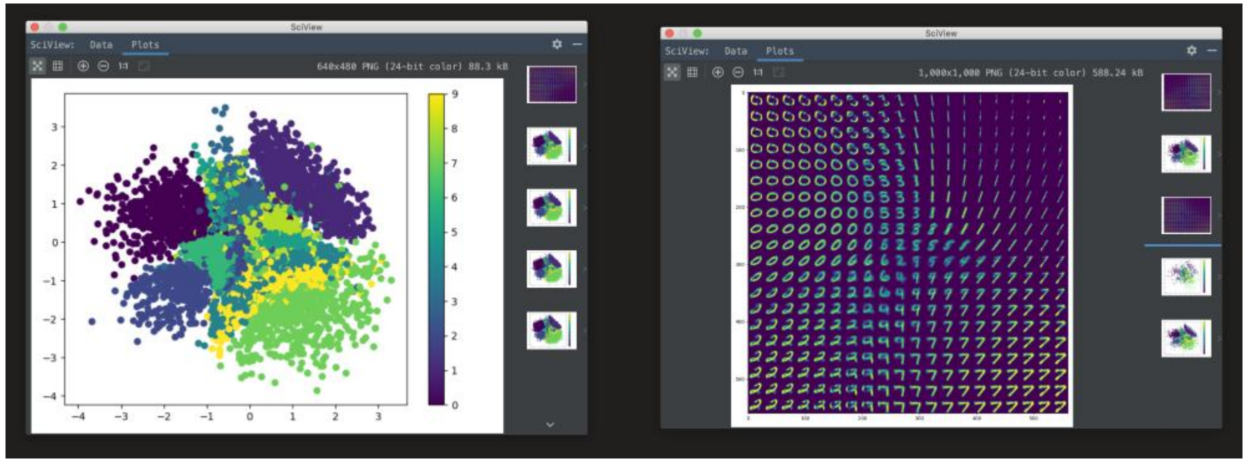


(b) Sampling

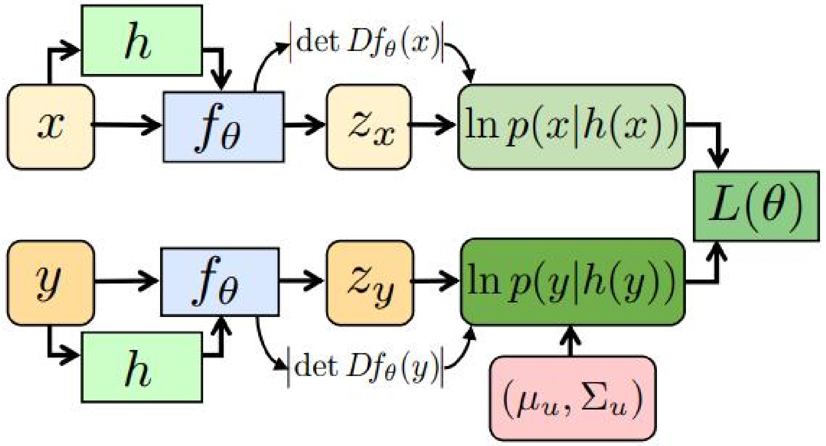
Model

DeFlow: Learning Complex Image Degradations from Unpaired Data with Conditional Flows

$$x \sim p_x = \mathcal{N}(\mu_x, \sigma_x^2)$$
$$y = x + \mu$$
$$\mu \sim p_\mu = \mathcal{N}(\mu_\mu, \sigma_\mu^2)$$
$$y \sim p_y = \mathcal{N}(\mu_x + \mu_\mu, \sigma_x^2 + \sigma_\mu^2)$$
$$\theta = \{\mu_x, \sigma_x^2, \mu_\mu, \sigma_\mu^2\}$$
$$L(\theta) = -\frac{1}{n}\sum_{i=1}^{n} lnp_x(x_i) - \frac{1}{m}\sum_{j=1}^{m} lnp_y(y_j)$$



$$x = f_\theta^{-1}(z_x)\,, \quad y = f_\theta^{-1}(z_y) = f_\theta^{-1}(z_x + u)$$
$$z_x \sim \mathcal{N}(0, I)\,, \quad u \sim p_u = \mathcal{N}(\mu_u, \Sigma_u)\,, \quad z_x \perp u$$

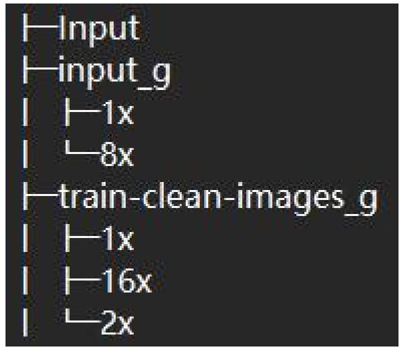


(a) Training

Model **DeFlow:** Learning Complex Image Degradations from Unpaired Data with Conditional Flows

1. Generate 2 different scales for dataset used to train Deflow.

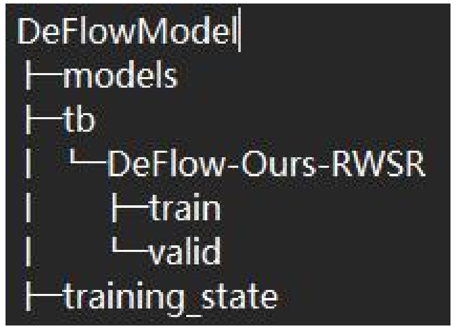
```
python create_DeFlow_train_dataset.py -source ./AIM-RWSR/input/ -target ./AIM-RWSR/input_g/ -scales 1 8
python create_DeFlow_train_dataset.py -source ./AIM-RWSR/train-clean-images/ -target ./AIM-RWSR/train-clean-images_g/ -scales 1  2  16
```



```
├─Input
├─input_g
|   ├─1x
|   └─8x
├─train-clean-images_g
|   ├─1x
|   ├─16x
|   └─2x
```

2. Train DeFlow

```
cd code
python train.py -opt ./confs/DeFlow-Our-RWSR.yml
```

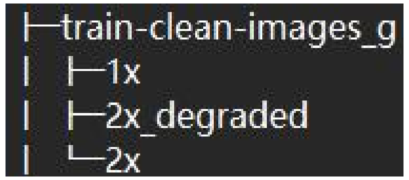


```
DeFlowModel
├─models
├─tb
|   └─DeFlow-Ours-RWSR
|       ├─train
|       └─valid
├─training_state
```

Model **DeFlow:** Learning Complex Image Degradations from Unpaired Data with Conditional Flows

3. DeFlow Inference on dataset DIV2K to generate degraded images.

```
python translate.py -opt DeFlow-Our-RWSR.yml -model_path ../trained_models/DeFlow_models/DeFlow-Ours-RWSR-100k.pth
\  -source_dir ../datasets/AIM-RWSR/train-clean-images_g/2x/ -out_dir ../datasets/AIM-RWSR/train-clean-images_g/2x_degraded/
```



```
├─train-clean-images_g
|   ├─1x
|   ├─2x_degraded
|   └─2x
```

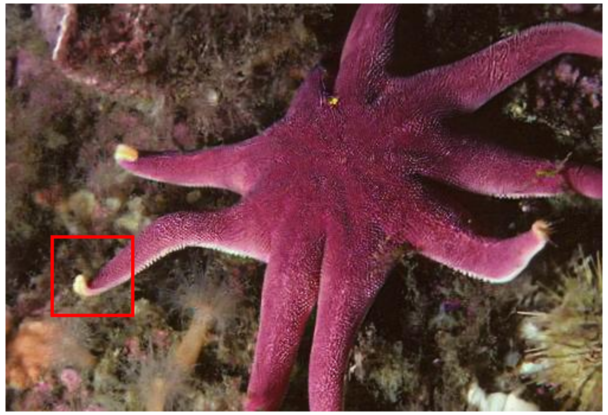
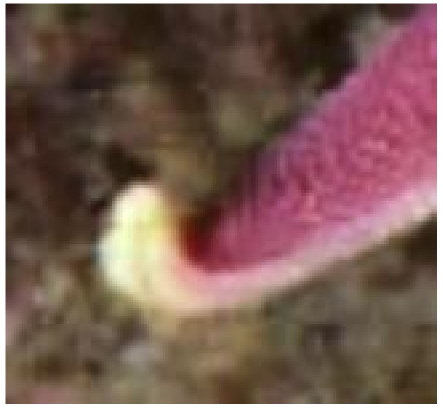
4. Use generated paired dataset to train ESRGAN.

```
cd ../Real-SR/codes/
python train.py -opt ../../ESRGAN_confs/train/ESRGAN-DeFlow-Ours-RWSR.yml
```

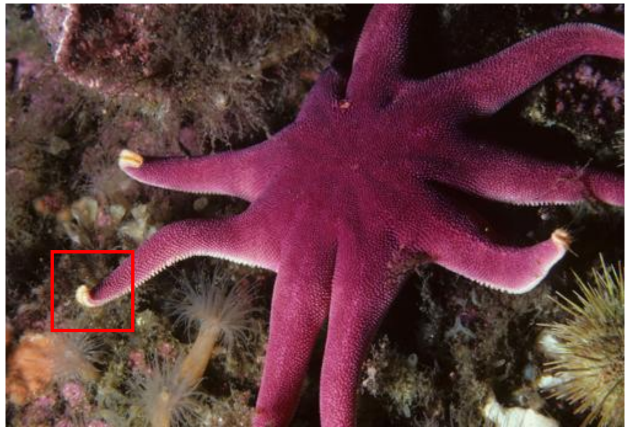
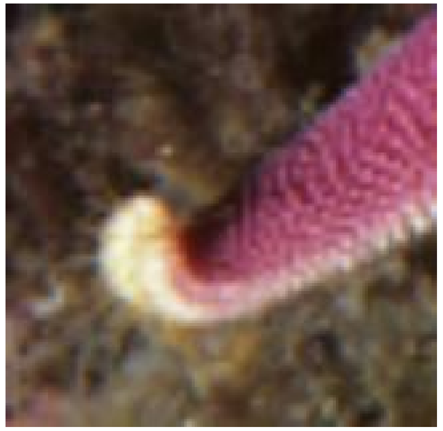
5. Use trained ESRGAN to validate on data in /input file and check the PSNR SSIM and LPIPS with the images in /GT file.

```
python test.py -opt ../../ESRGAN_confs/test/ESRGAN-DeFlow-Ours-RWSR.yml
```

Degraded Paired Data of Task 1

Degraded

GT

Metrics

1. PSNR

The PSNR is defined through MSE:

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

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

Metrics

2. SSIM

The SSIM index is calculated on various windows of an image. The measure between two windows x and y of common size N x N is:

$$\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

with:

μ_x the average of x;
μ_y the average of y;
σ_x^2 the variance of x;
σ_y^2 the variance of y;
σ_{xy} the covariance of x and y;
$c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ two variables to stabilize the division with weak denominator;
L the dynamic range of the pixel-values;
$k_1 = 0.01$ and $k_2 = 0.03$ by default.

Metrics

3. LPIPS[1]

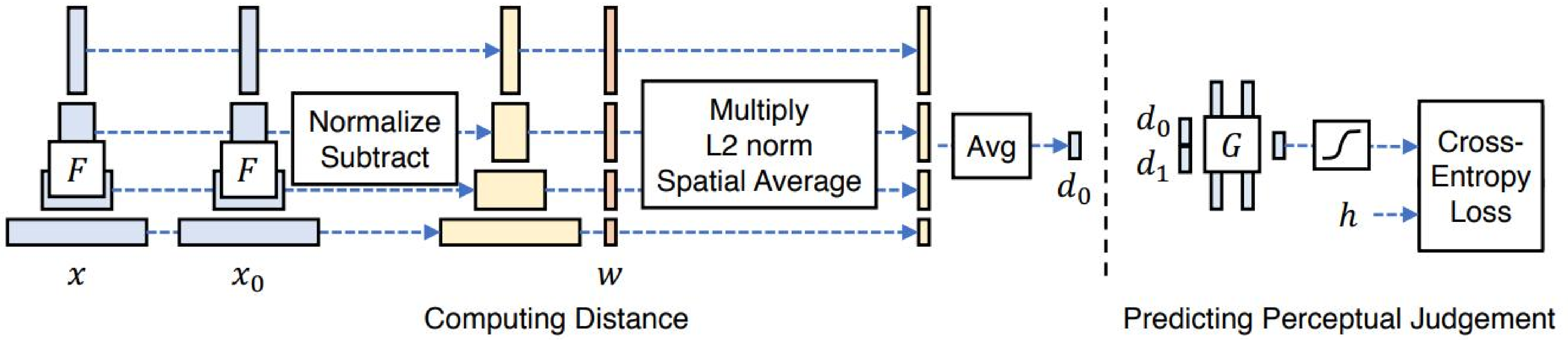


Figure 3: **Computing distance from a network** (Left) To compute a distance d_0 between two patches, x, x_0, given a network $\mathcal{F}$, we first compute deep embeddings, normalize the activations in the channel dimension, scale each channel by vector w, and take the ℓ_2 distance. We then average across spatial dimension and across all layers. (Right) A small network $\mathcal{G}$ is trained to predict perceptual judgment h from distance pair (d_0, d_1).

Learned Perceptual Image Patch Similarity (LPIPS),
a reference-based image quality metric based on feature distances in CNNs

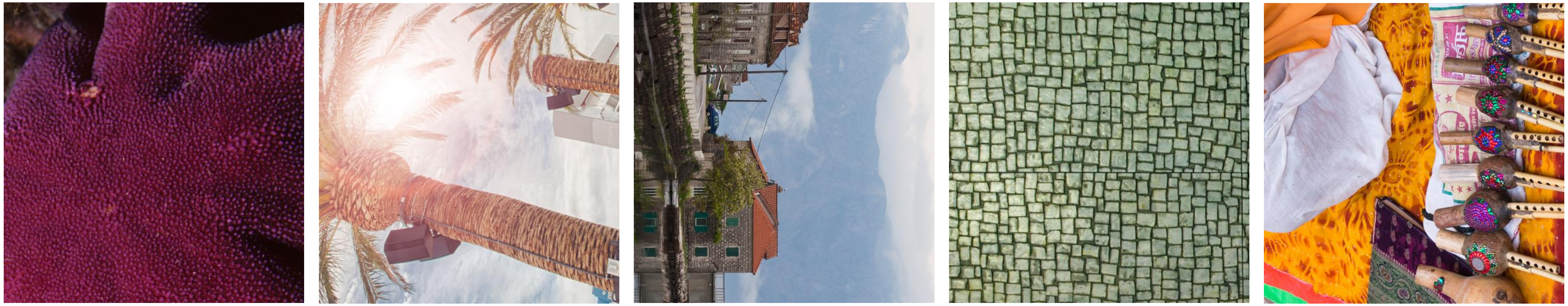
[1] Zhang R, Isola P, Efros A A, et al. The unreasonable effectiveness of deep features as a perceptual metric Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 586-595.

Result - Task 1

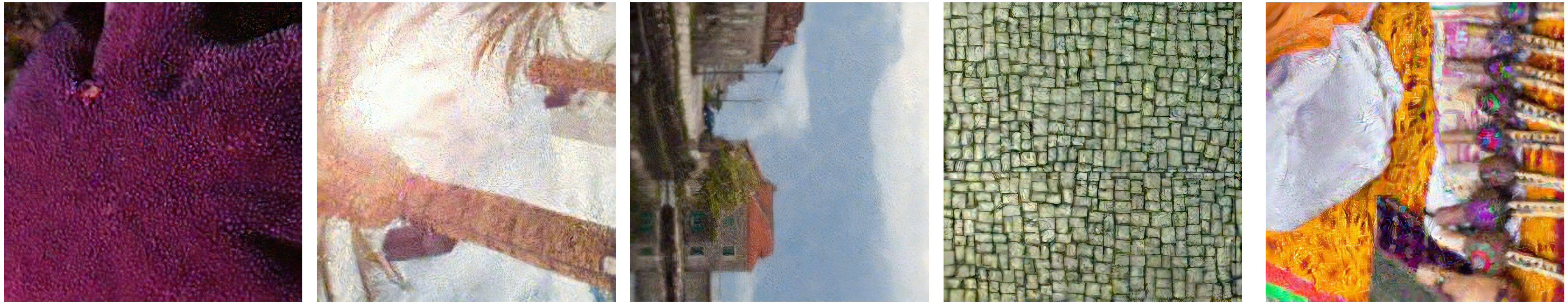
Overall performance for all generated images from task 1:

mean PSNR: 19.67
mean LPIPS: 0.3107
mean SSIM: 0.3666

Result - Task 1



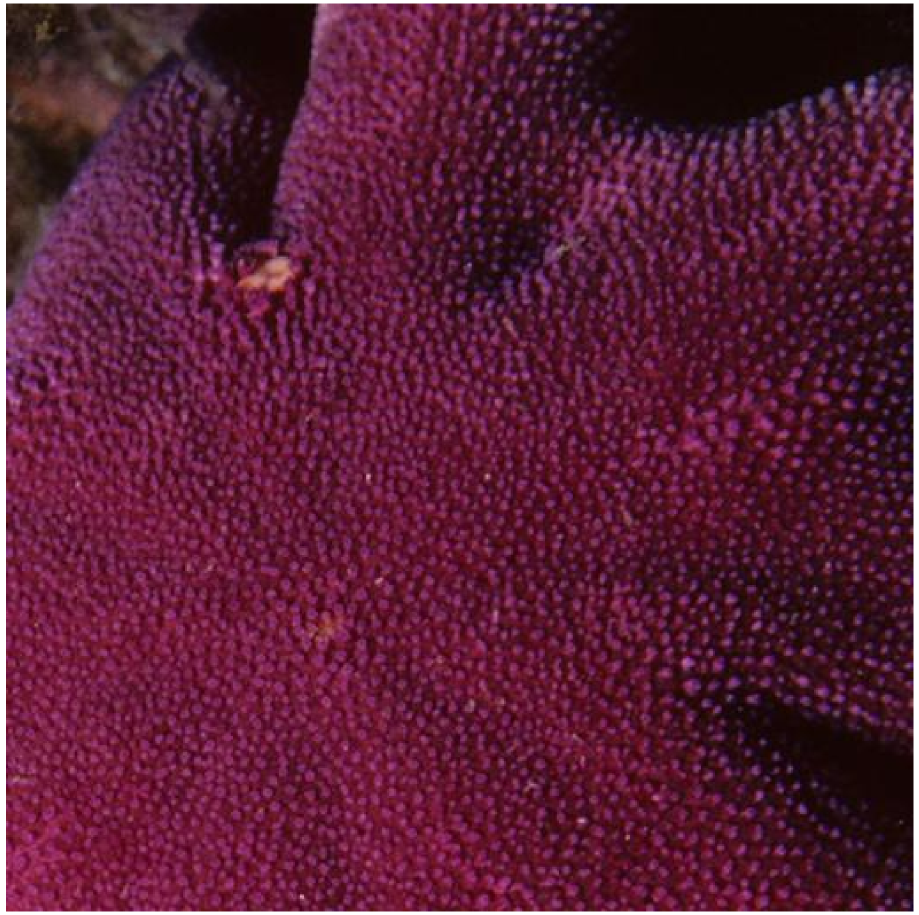
GT

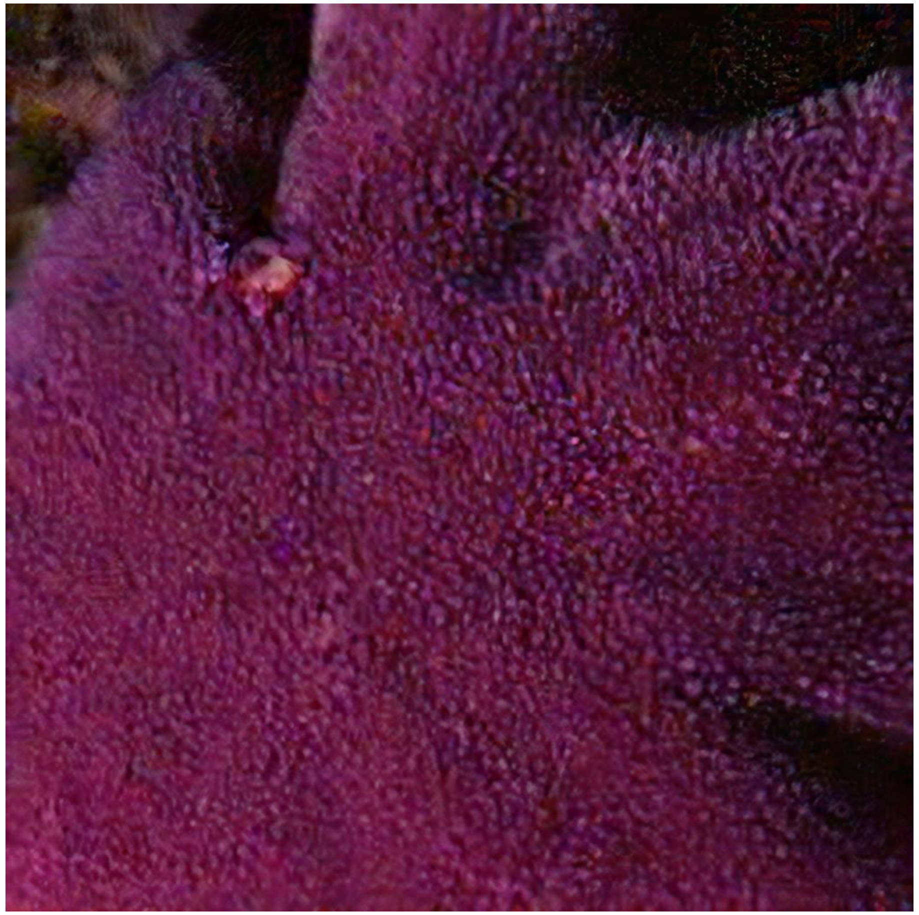


Defow + ESRGAN

PSNR: 21.13 PSNR: 18.80 PSNR: 16.50 PSNR: 20.28 PSNR: 17.61

Result - Task 1


GT


HR

Domain Invariant Mapping Function h

1. Domain Invariant Mapping Function h:

$$h(x) = d_{\downarrow}(x) + n,\ n \sim N(0, \sigma^2)$$

where $d_{\downarrow}(x)$ denotes bicubic downsampling.

2. How to design the new Domain Invariant Mapping Function h:

The purpose of h is to remove the original degradation, while preserving image content. Since Function h is to make the model cannot tell which domain the image is in, so:

(1) Add noise to both input images.
(2) Denoise for both input images.

Domain Invariant Mapping Function h

3.Noise Model:

There are two kinds of Noise Models:

(1)Additive Noise Model

$$w(x, y) = s(x, y) + n(x, y)$$

(2)Multiplicative Noise Model

$$w(x, y) = s(x, y) \times n(x, y)$$

Domain Invariant Mapping Function h

4. Denoise

Spatial domain filtering

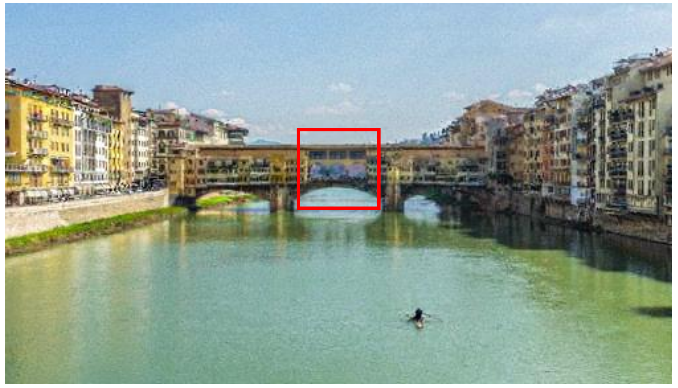
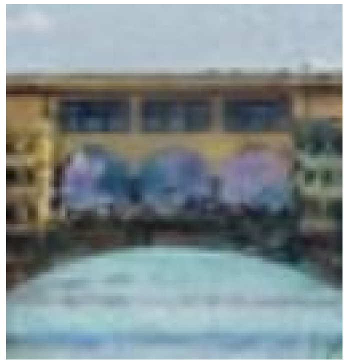
(1) Smoothing spatial filter
(2) Median filter
(3) Bilateral filter

5. Down Sampling and Add Noise

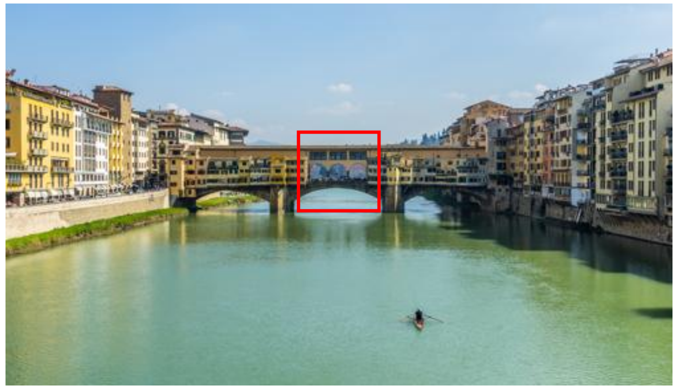
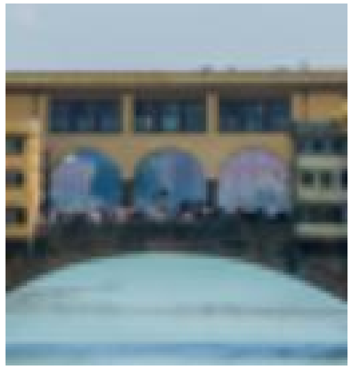
Use Maxpooling instead of bicubic downsampling.

6. Denoising Autoencoder

Degraded Paired Data of Task 2

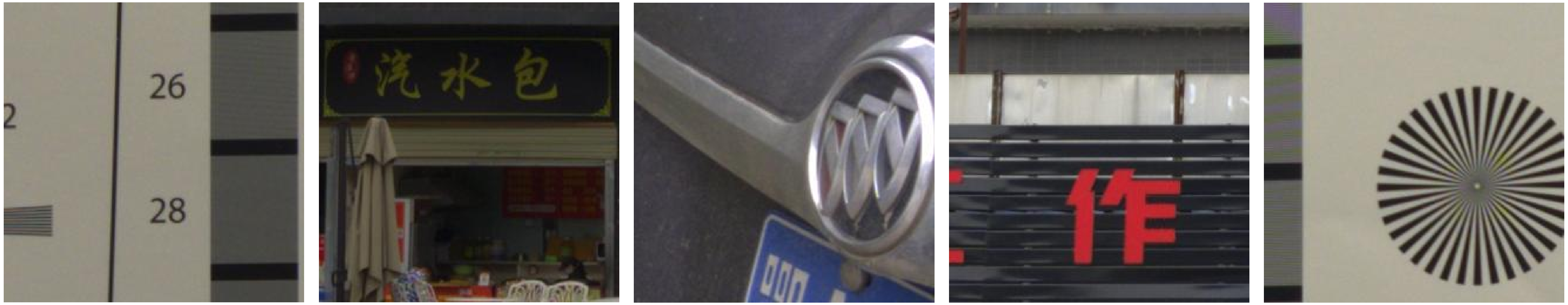



Degraded

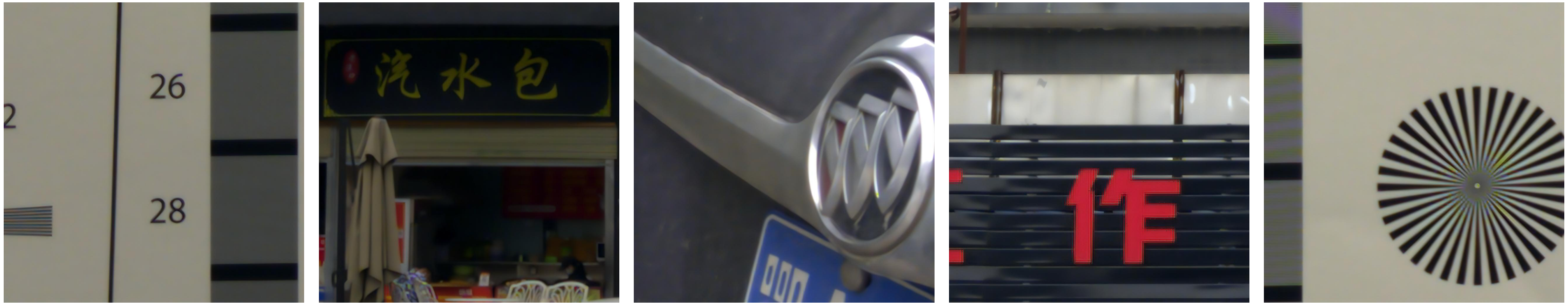



GT

Result - Task 2

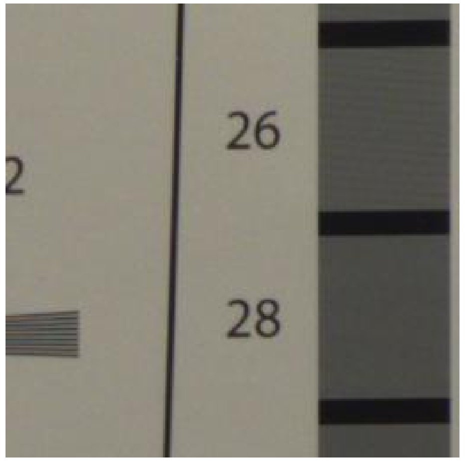


LR

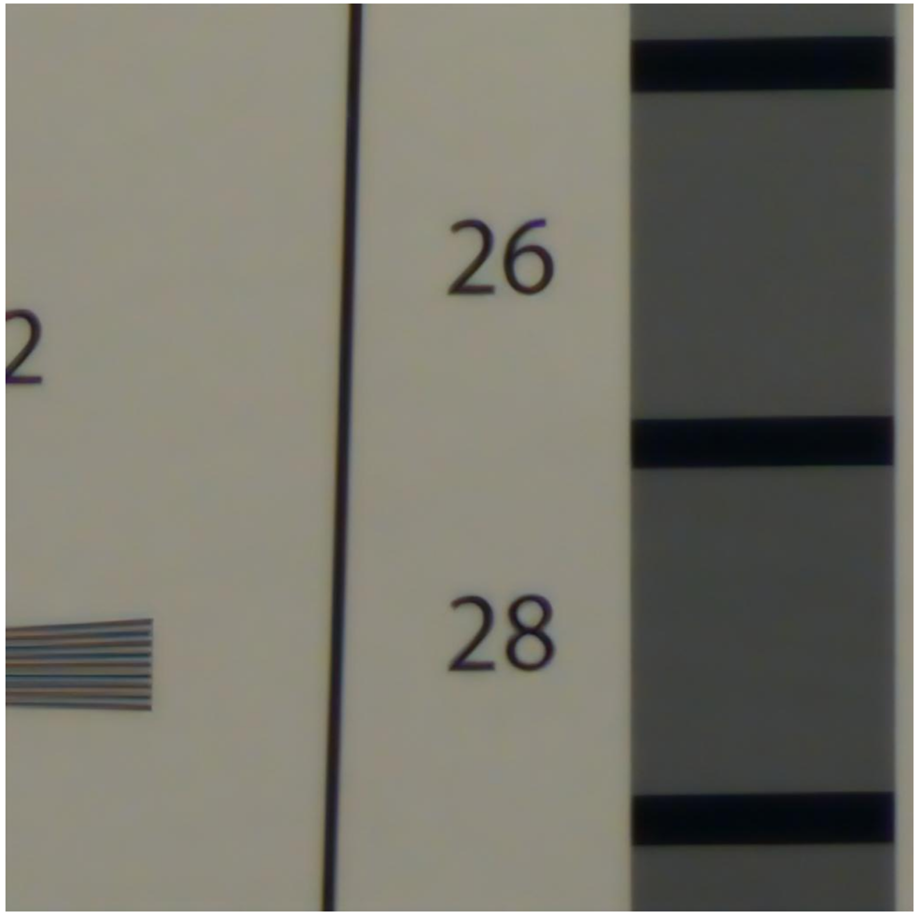


HR

Result - Task 2

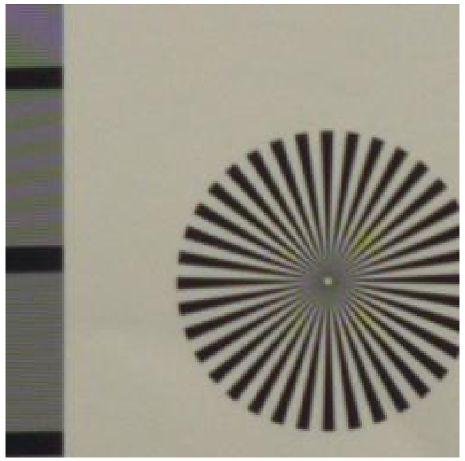


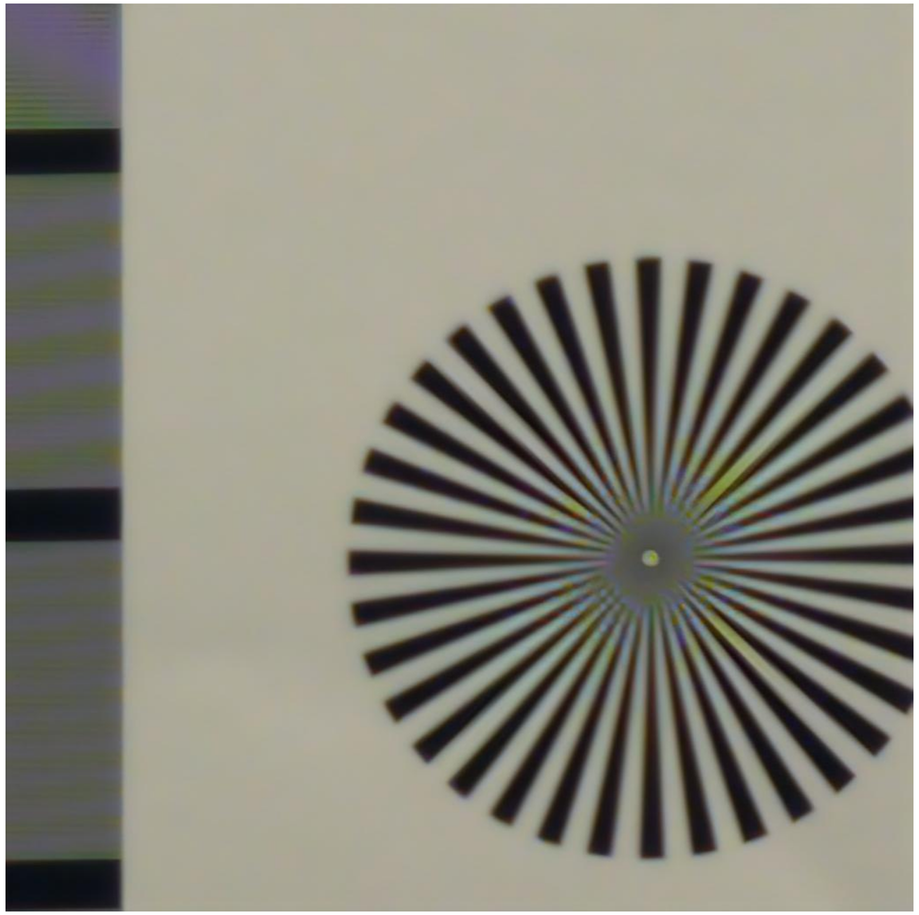
LR



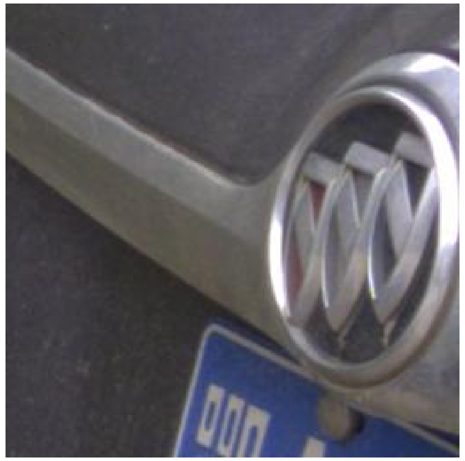
HR

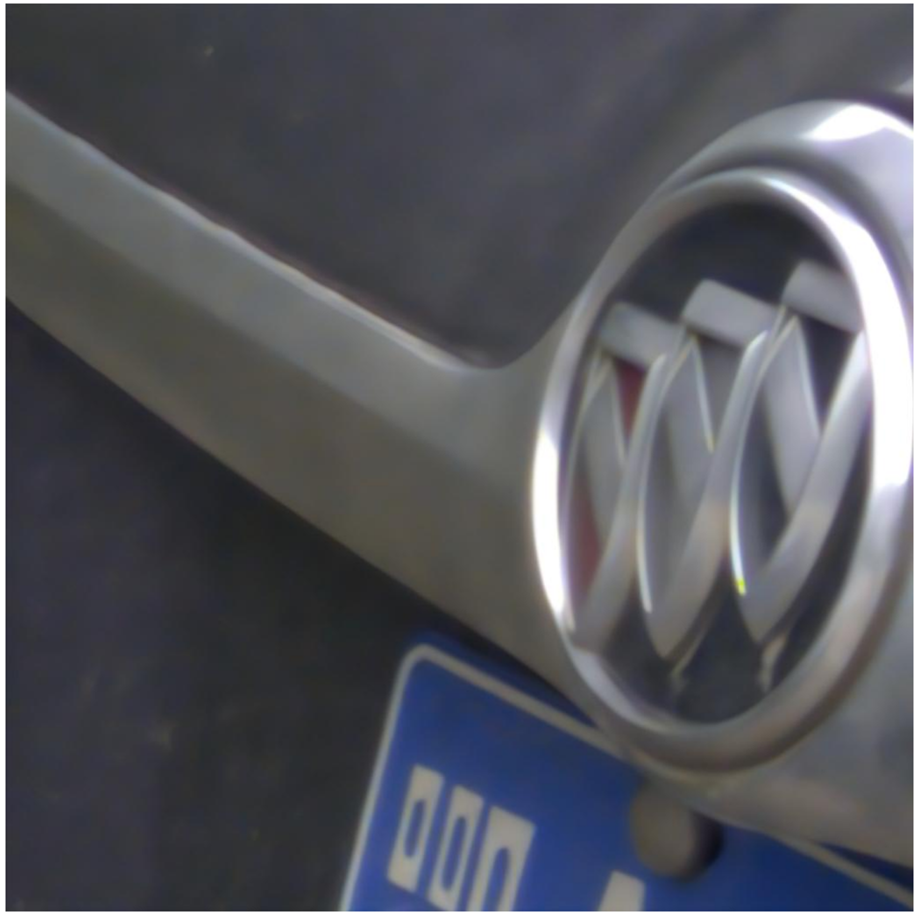
Result - Task 2


LR


HR

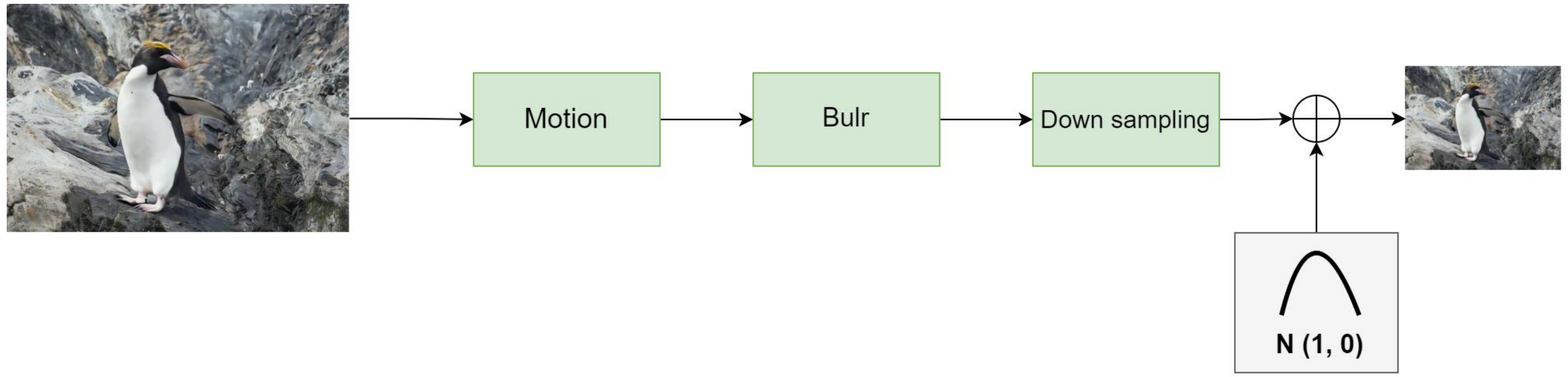
Result - Task 2


LR


HR

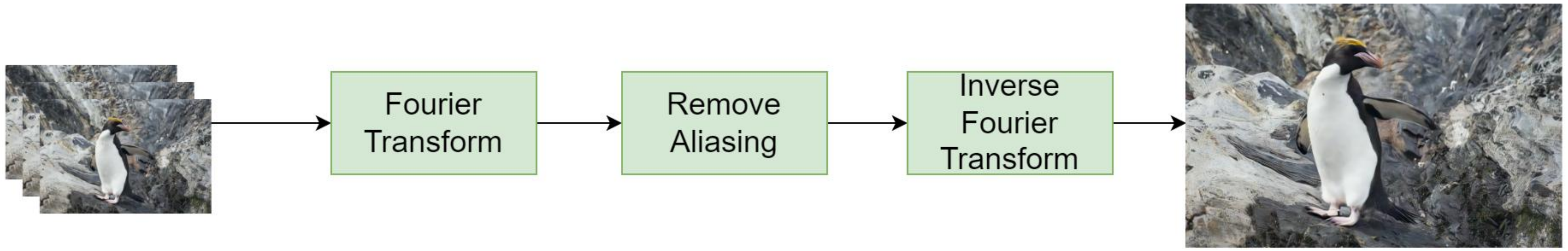
Traditional Method vs Deep Learning

1. General noisy model



2. Traditional SR models:

2.1 Frequency Domian



Traditional Method vs Deep Learning

2.1 The methods based on frequency domain

The methods based on frequency domain are mainly Fourier transform and its inverse transform. Since image details are reflected by high-frequency information, eliminating spectral aliasing in low-resolution images can obtain more masked high-frequency information, thereby increasing image details and improving image resolution.

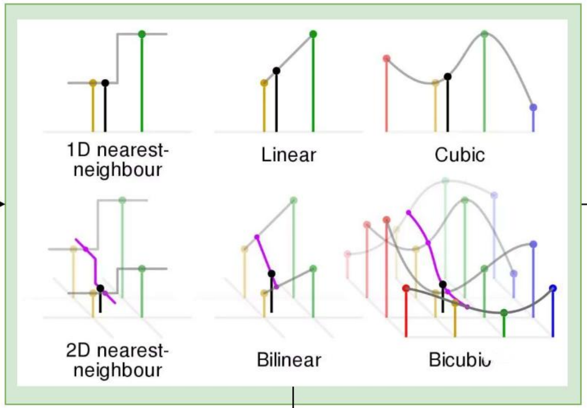
Advantages: clear principle, convenient theoretical derivation, low computational complexity;

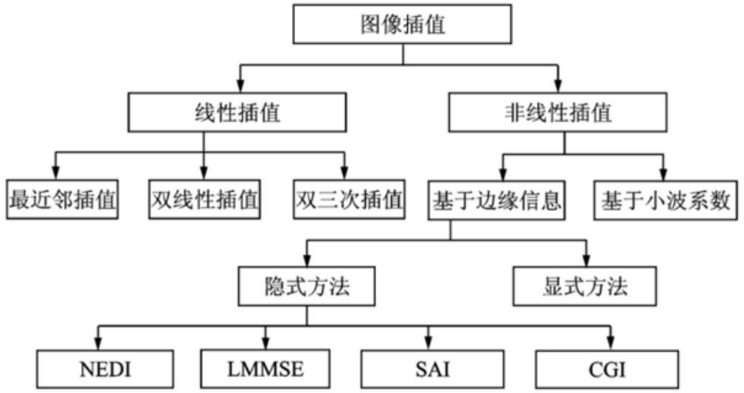
Disadvantages: It is only suitable for the case of spatially invariant noise, and it can only deal with the situation where there is only global motion in the image without local motion, and it is difficult to use prior information in the processing process.

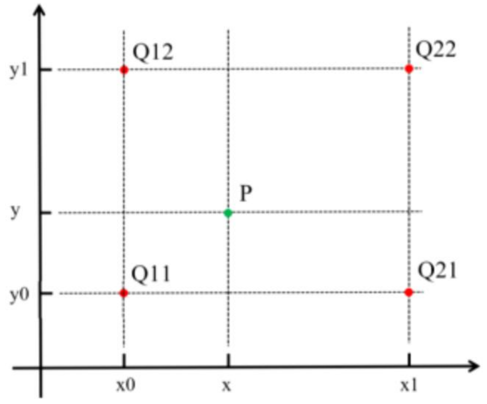
(

Traditional Method vs Deep Learning

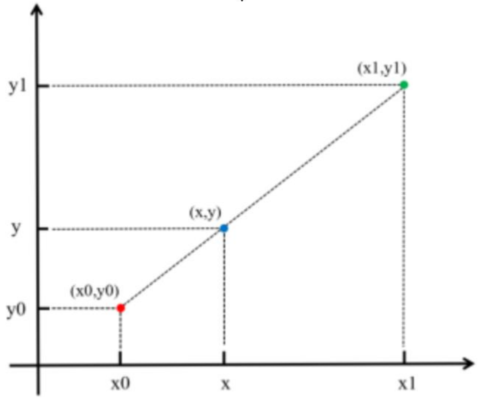
2.2 Space Domain



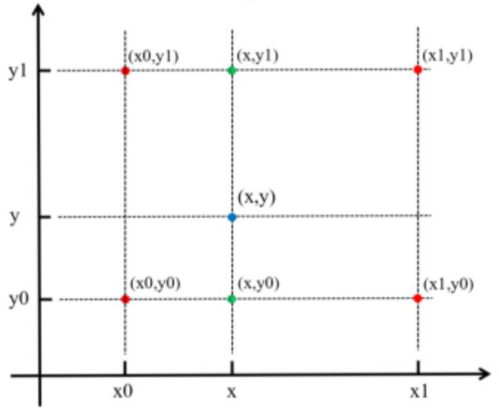




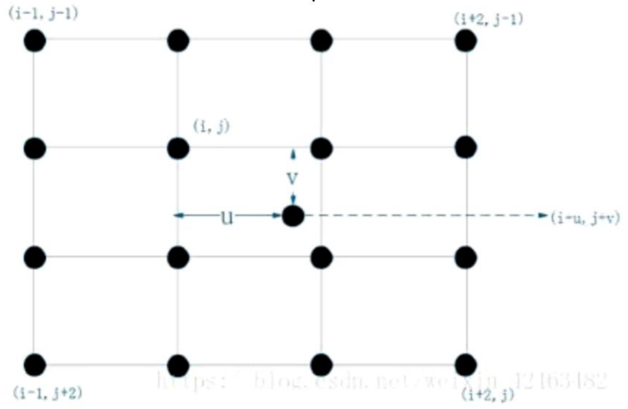
Nearest Neighbor Interpolation



Linear Interpolation



Bilinear Interpolation



Bicubic Interpolation

Traditional Method vs Deep Learning

2.2 Method based on space domian

The method of airspace often uses the local information of the image to increase the number and density of pixels, thereby increasing the details of the image and improving the resolution of the image.

Advantages: There are many types, various degradation factors can be considered comprehensively, flexibility;

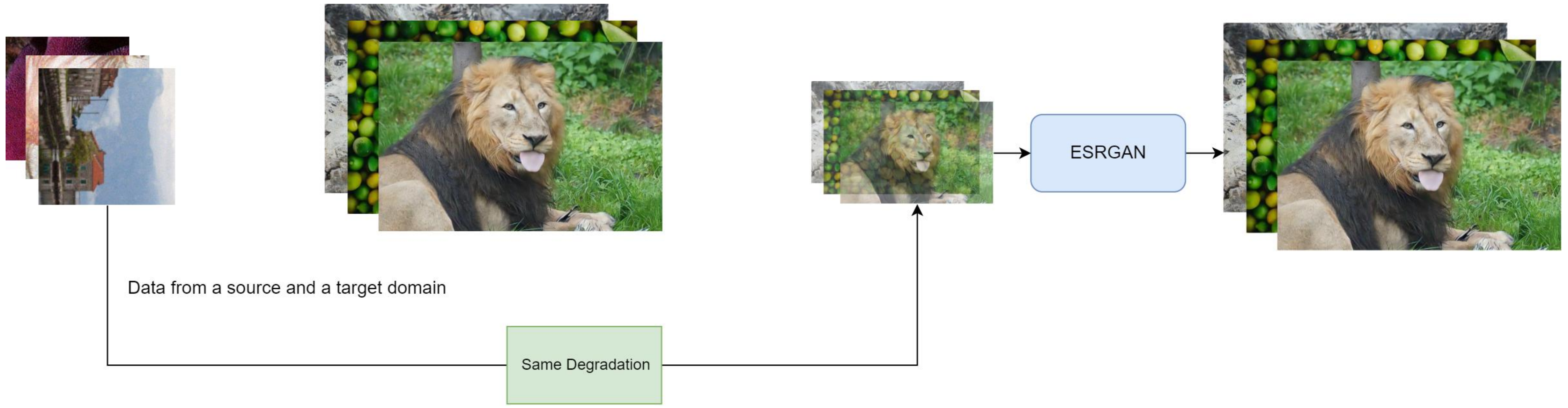
Disadvantages: complex design and high computational complexity.

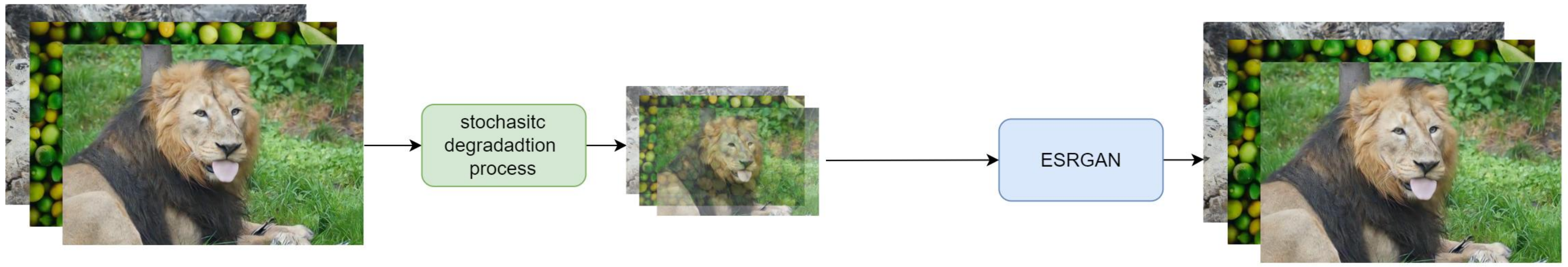
Combing both deep-learning and traditional method

In this SR task, we used deep-learning in both paired data generation and HR image generation.

If tradition method is considered. We should try using traditional method to replace paired data generation model to have a Lightweight model.

Combing both deep-learning and traditional method

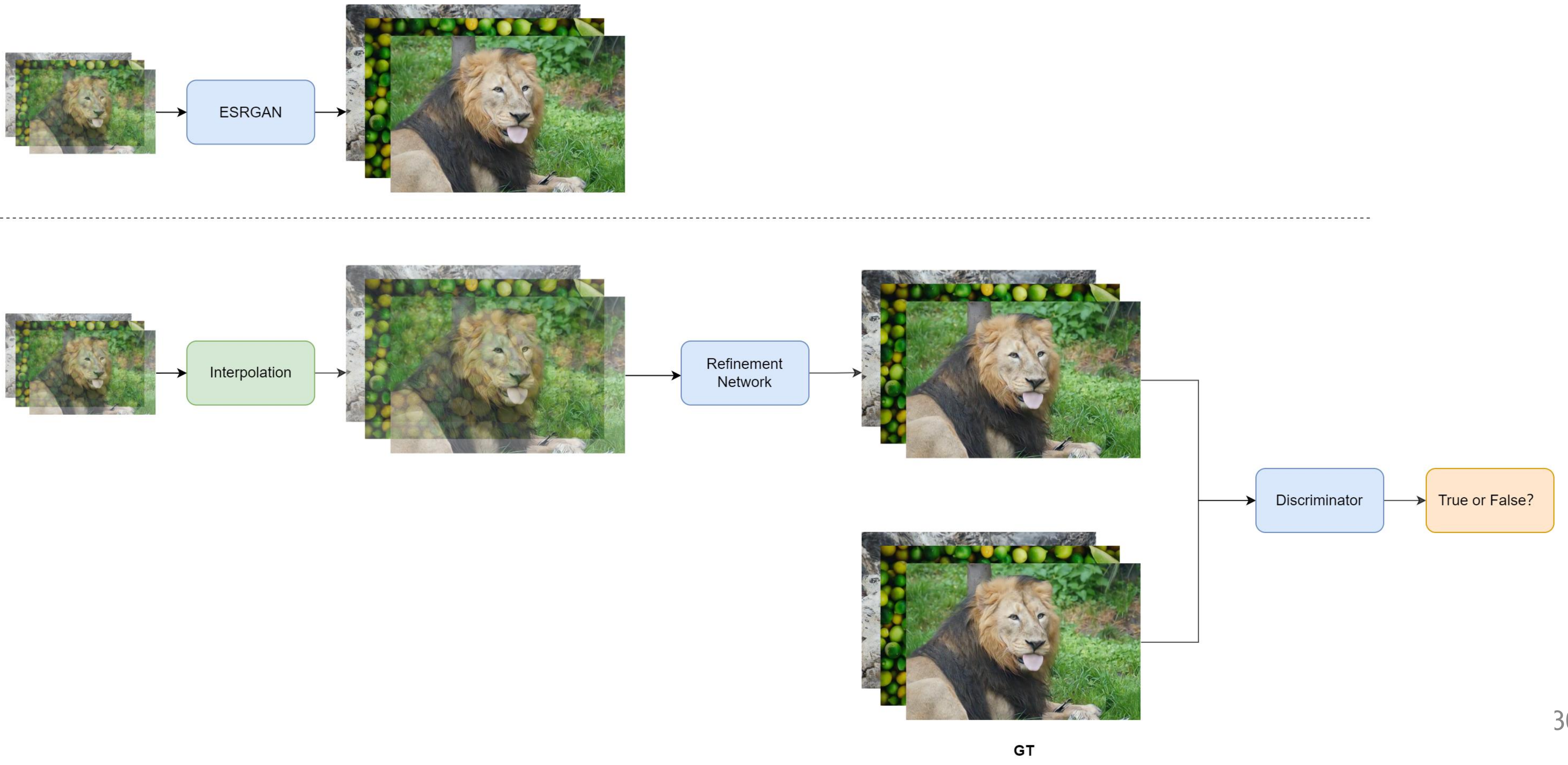




Tradition methods in Frequency and Space Domain

Combing both deep-learning and traditional method

Instead of using GAN model to generate HR images from LR images(up), we could use a Refinement Network to refine the image generated by traditional method(bottom). This could reduce the calcualtion and also with good performance.



Thank you very much!!

Q&A