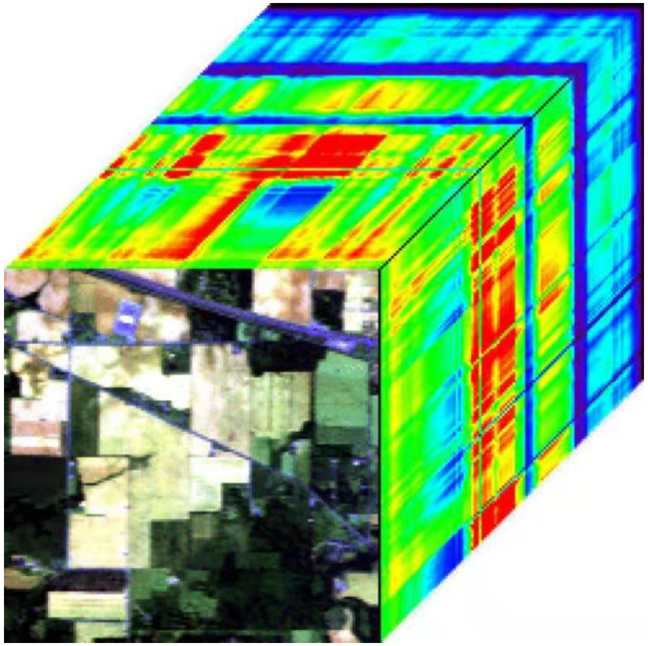
ADRN:Attention-Based Deep Residual Network for Hyperspectral Image Denoising

Yongsen Zhao, Deming Zhai, Junjun Jiang, Xianming Liu

Harbin Institute of Technology, Harbin, China, 150001

ICASSP 2020

Background



- abundant spatial and spectral information
- Washington DC Mall image, 191 bands

Background

- The goal of HSI denoising is to recover a clean image $\mathbf{x}$ from a noisy observation $\mathbf{y}$,

$$\mathbf{y} = \mathbf{x} + \mathbf{v}$$

- where $\mathbf{v}$ is additive white Gaussian noise in general.
- To address this ill-posed inverse problem, some ***prior knowledge*** about $\mathbf{x}$ needs to be adopted.

Background

- Non-local
  - BM4D
- Low-rank
  - LRTA, LRMR, LLRT
- Non-local and Low-rank
  - NG-Meet

> ***Time-consuming***
>
> ***Prior is hand-craft and thus lack of representation ability***

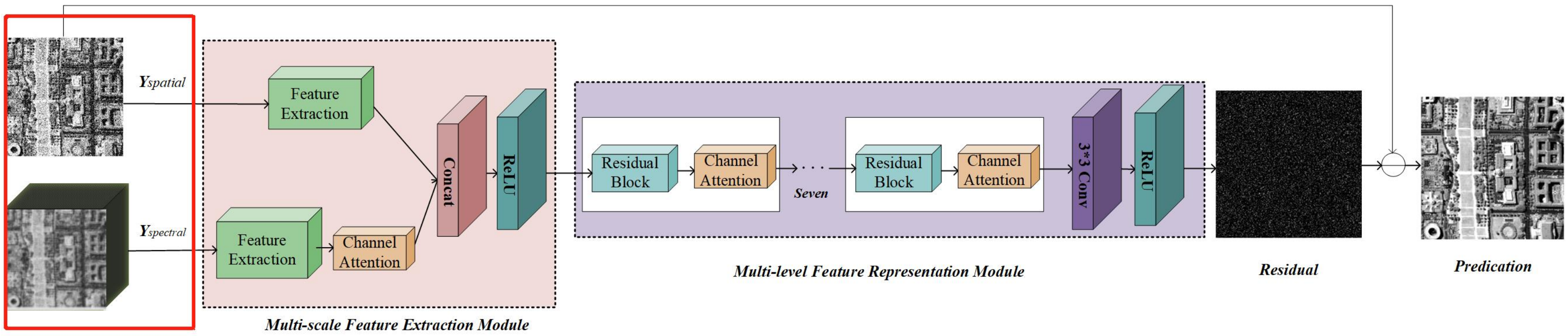
Background

- Deep-learning based method
    - HSID-CNN, SSGN

> ***How to better capture both the spatial and spectral information?***

> ***How to design more discriminate network structure and improve the representation ability?***

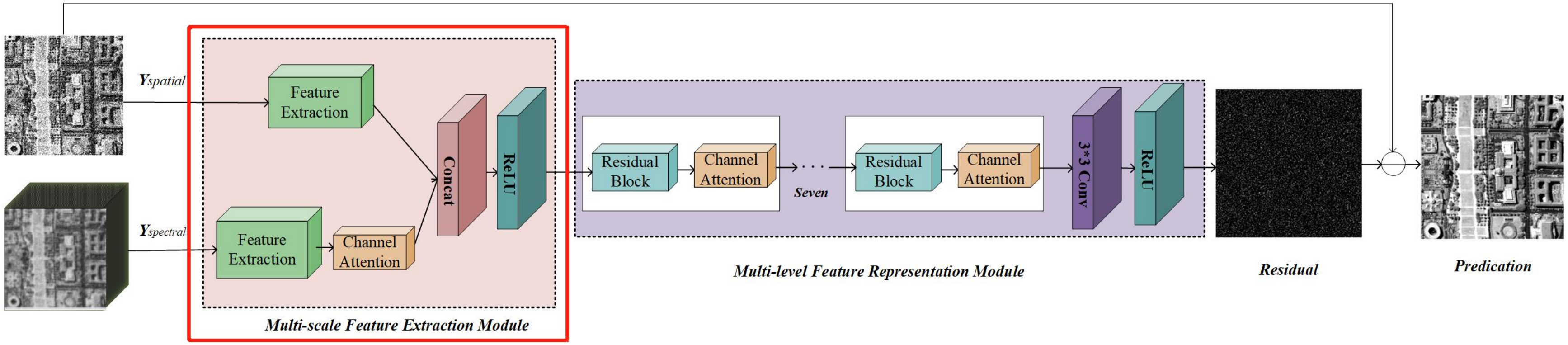
Methodology



Overall architecture

- $Y_{spatial}$ denotes an input noisy band
- $Y_{spectral}$ denotes its K adjacent bands
- use auxiliary input to capture the low-rank property

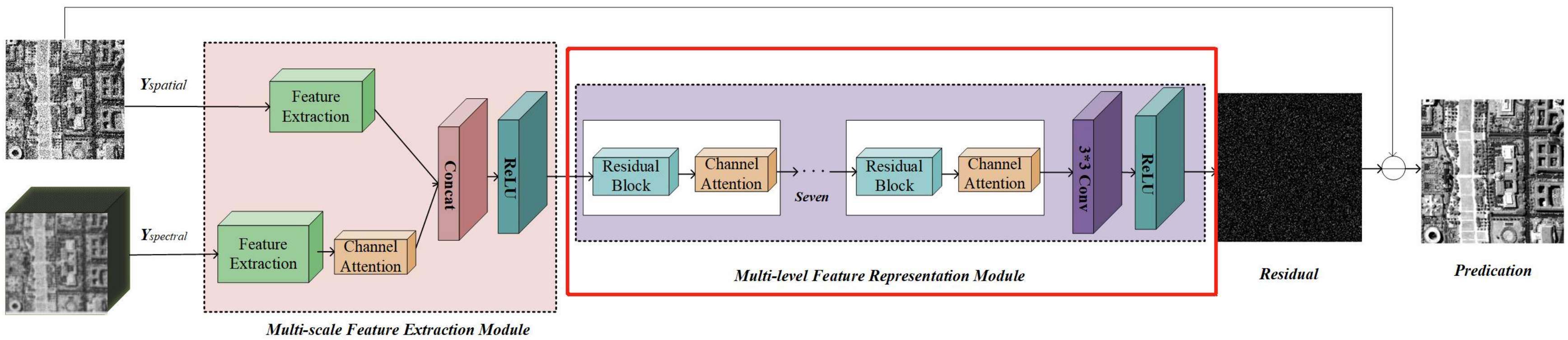
Methodology



Overall architecture

- extract the multi-scale spatial and spectral information
- concentrate on the most relevant feature

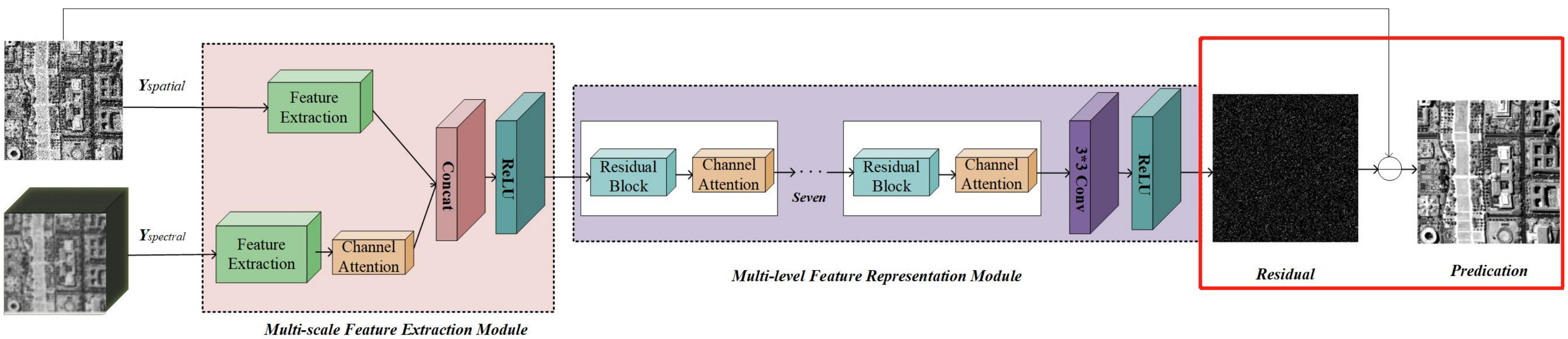
Methodology



Overall architecture

- fuse the multi-level feature
- construct the residual noise

Methodology

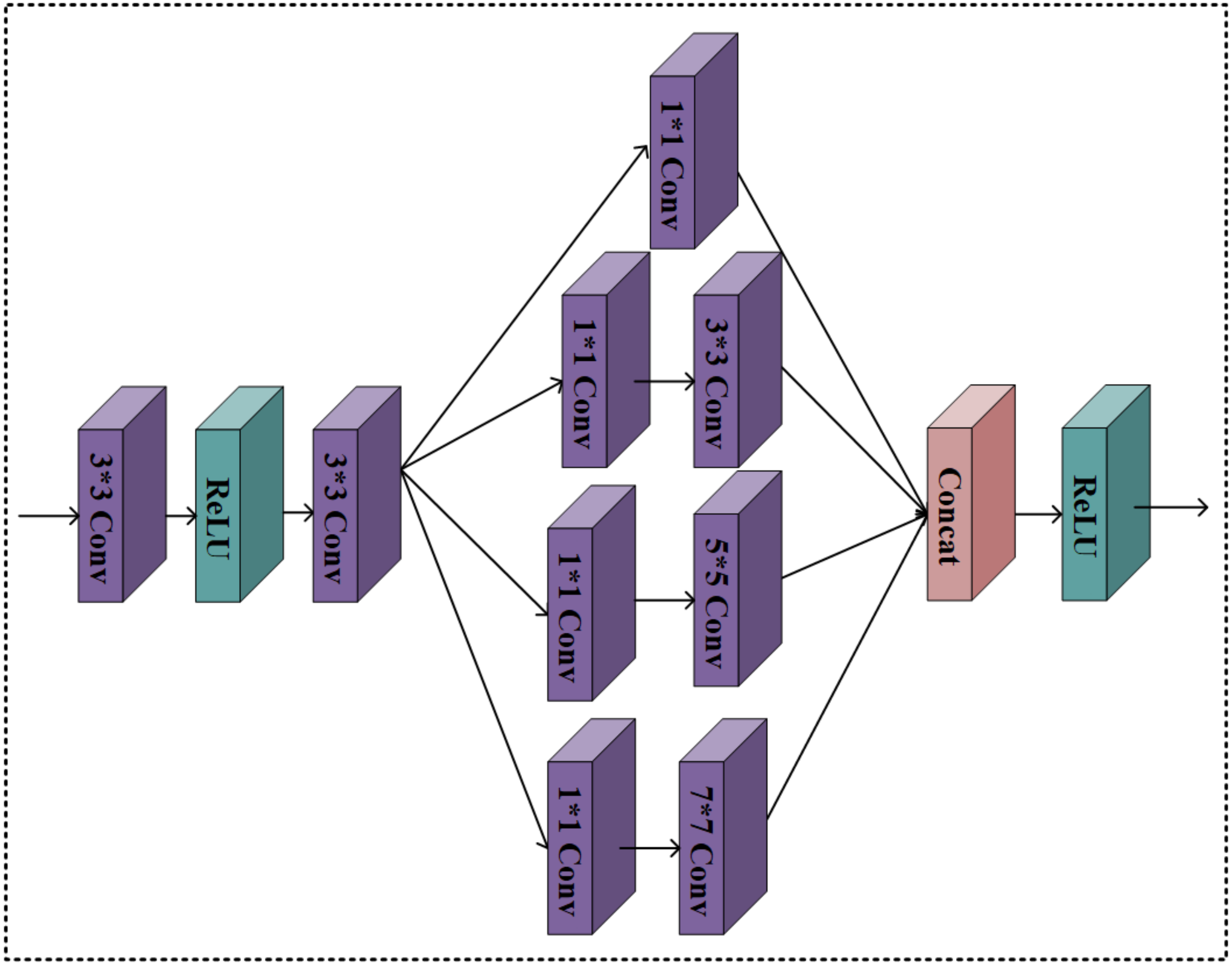


Overall architecture

$$R = F(\Theta, Y_{spatial}, Y_{spectral})$$

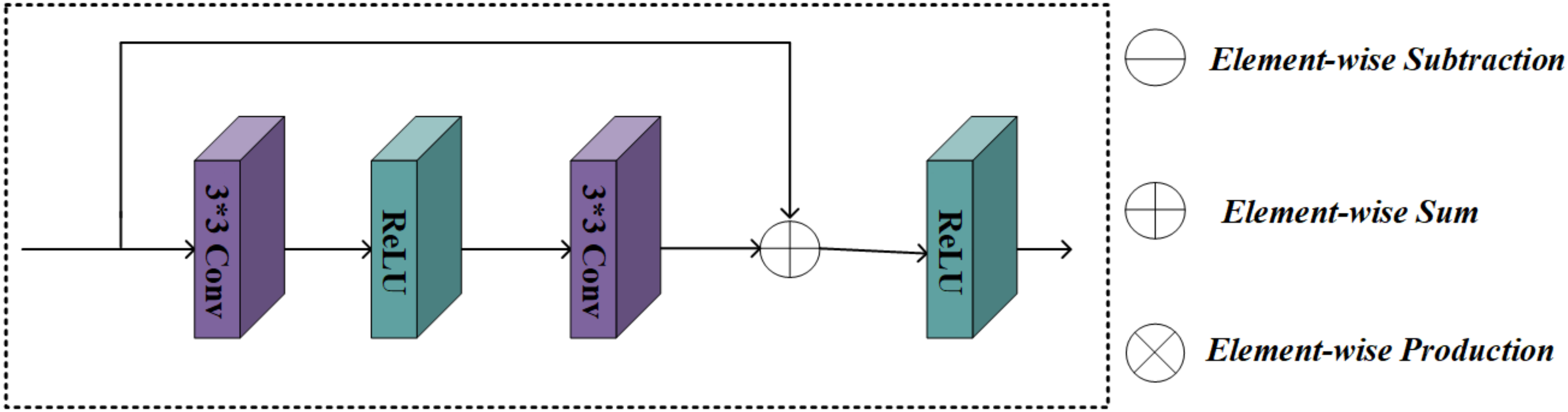
$$\hat{X} = Y_{spatial} - R$$

Methodology



Feature extraction block

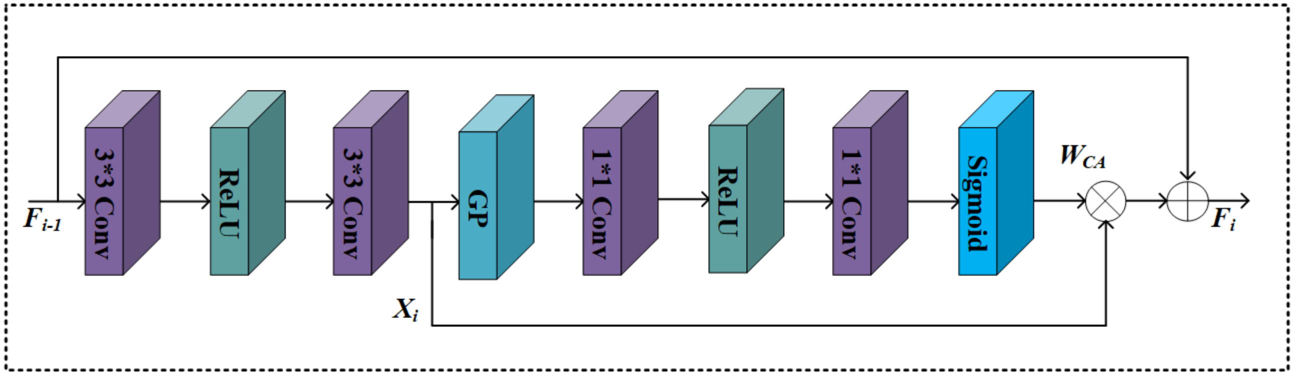
Methodology



Residual block

- increases the flow of information
- contribute to noise prediction and back propagation

Methodology



Channel attention block

$$F_i = F_{i-1} + W_{CA} * X_i$$

$$X_i = W_2 * \delta(W_1 * F_{i-1})$$

$$W_{CA} = Sigmoid(W_4 * \delta(W_3 * GP(X_i)))$$

- adaptively modulate feature representation

Methodology

- The loss function of our training process consists of two parts:

$$L_{total} = \lambda L_{rec} + L_{reg}$$

- L_{rec} aims to ensure the restored result approximate to the ground truth:

$$L_{rec} = \frac{1}{NHW} \sum_{i=1}^{N} ||\hat{X}^i - X^i||_2^2$$

- while L_{reg} is used to enforce the residual noise satisfy a zero-mean distribution:

$$L_{reg} = (\frac{1}{NHW} \sum_{i=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} R_{hw}^i)^2$$

Experiments and results

- Washington DC Mall image: $1280 \times 303 \times 191$
- Normalize the gray values of each HSI band to [0,1] using ENVI software
- Select the middle 200×200 for testing
- Crop 20×20 patches from the remaining part and impose additive white Gaussian noise to formulate the training data

Experiments and results

> three types of noise are employed during test

- different bands have the same noise intensity, σ_n is set from 5 to 100
- the noise intensity of different bands conforms a random probability distribution, labeled as *rand(25)*
- for different bands, the noise intensity is also different but varies like a Gaussian distribution centered at the middle band, labeled as *Gau(200， 30)*

$$\sigma_n = \beta \sqrt{\frac{exp\{-(k-B/2)^2/2\eta^2\}}{\sum_{k=1}^{B} exp\{-(k-B/2)^2/2\eta^2\}}}$$

Experiments and results

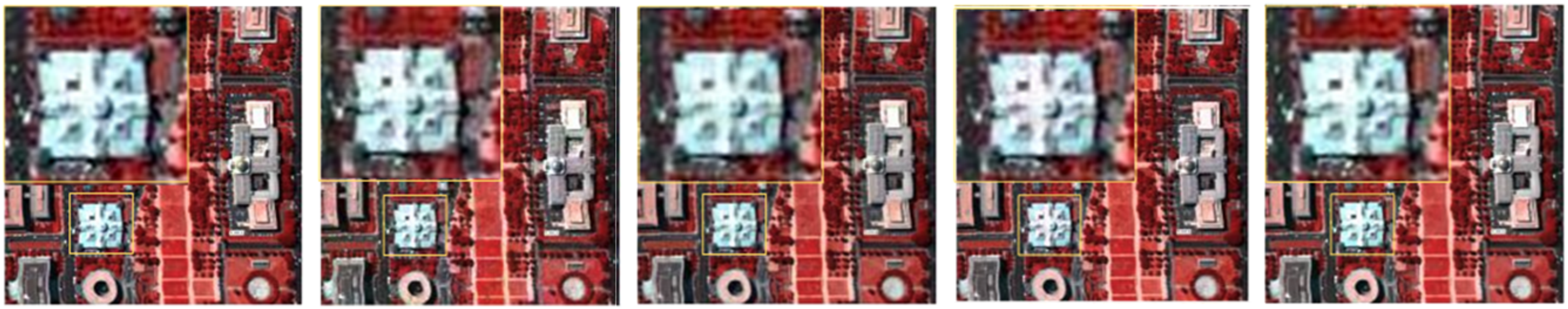
- K = 64, $\lambda = 10$
- Adam with a batchsize of 382
- Use the truncated normal distribution to initialize the weights
- Learning rate starts from 1e-4 and decays exponentially every certain training steps (such as 5000)
- Roughly 300,000 iterations

Experiments and results

Table 1. Quantitative performance comparison of the denoising results

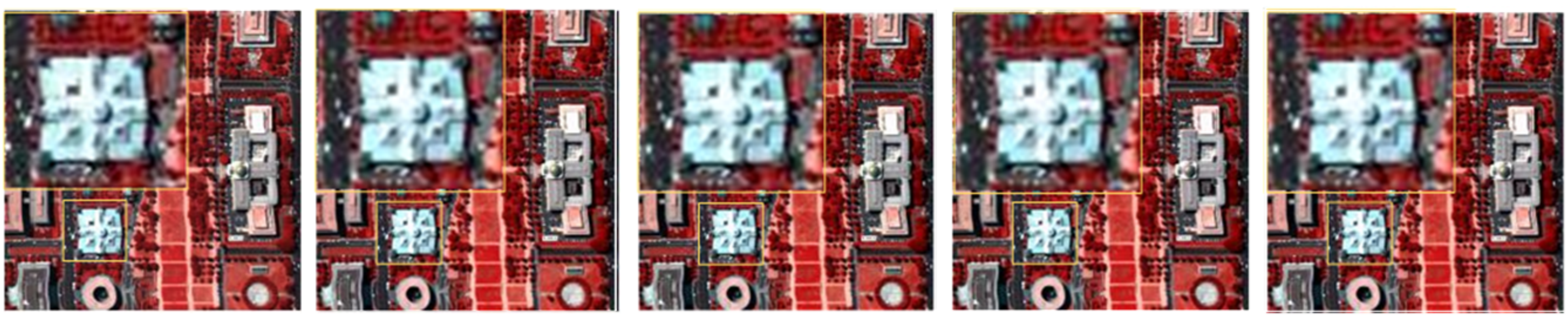
Noise Level	Criterion	LRTA [3]	BM4D [2]	LRMR [4]	HSID-CNN [7]	LLRT [5]	NG-Meet [6]	Proposed
$\sigma_n = 5$	MPSNR	39.009±0.0034	41.188±0.0023	40.878±0.0036	41.684±0.0025	41.532±0.0054	**41.781±0.0052**	41.580±0.0043
	MSSIM	0.9926±0.0002	0.9962±0.0001	0.9952±0.0001	0.9966±0.0001	0.9968±0.0001	0.9966±0.0001	**0.9972±0.0001**
$\sigma_n = 25$	MPSNR	30.672±0.0033	31.136±0.0025	33.029±0.0023	33.050±0.0028	34.701±0.0097	35.366±0.0094	**35.527±0.0104**
	MSSIM	0.9629±0.0002	0.9685±0.0002	0.9809±0.0001	0.9813±0.0001	0.9862±0.0 001	0.9880±0.0001	**0.9902±0.0001**
$\sigma_n = 50$	MPSNR	26.832±0.0052	26.752±0.0034	28.806±0.0043	28.968±0.0039	30.759±0.0115	31.669±0.0139	**32.070±0.0102**
	MSSIM	0.9246±0.0001	0.9208±0.0002	0.9532±0.0001	0.9536±0.0001	0.9705±0.0001	0.9752±0.0001	**0.9796±0.0001**
$\sigma_n = 75$	MPSNR	24.682±0.0054	24.261±0.0035	26.306±0.0046	26.753±0.0039	28.385±0.0134	29.116±0.0147	**29.862±0.0175**
	MSSIM	0.8866±0.0001	0.8670±0.0001	0.9192±0.0001	0.9273±0.0001	0.9525±0.0002	0.9594±0.0001	**0.9673±0.0001**
$\sigma_n = 100$	MPSNR	23.175±0.0048	22.577±0.0054	24.310±0.0047	25.296±0.0043	26.712±0.0145	27.756±0.0083	**28.239±0.0176**
	MSSIM	0.8494±0.0003	0.8119±0.0002	0.8799±0.0002	0.9014±0.0001	0.9328±0.0001	0.9454±0.0001	**0.9535±0.0002**
$\sigma_n = rand(25)$	MPSNR	28.843±0.0025	34.424±0.0034	36.094±0.0033	37.367±0.0028	34.360±2.6908	36.040±0.3682	**37.301±0.1633**
	MSSIM	0.9331±0.0001	0.9833±0.0002	0.9856±0.0001	0.9916±0.0001	0.9718±0.0275	0.9904±0.0001	**0.9917±0.0004**
$\sigma_n = Gau(200, 30)$	MPSNR	28.200±0.0023	34.109±0.0037	35.962±0.0025	36.804±0.0029	28.635±0.0019	35.402±0.0053	**37.722±0.0080**
	MSSIM	0.9119±0.0002	0.9794±0.0001	0.9893±0.0001	0.9895±0.0001	0.9094±0.000	0.9894±0.0001	**0.9929±0.0001**

Experiments and results



(a) Clean image (b) HSID-CNN 25.635/0.9235 (c) LLRT 26.644/0.9287 (d) NG-Meet 27.667/0.9442 (e) Proposed 28.233/0.9536

$$\sigma_n = 100, Washington\ DC\ Mall$$



(a) Clean image (b) HSID-CNN 36.295/0.9927 (c) LLRT 33.857/0.9764 (d) NG-Meet 36.434/0.9912 (e) Proposed 37.333/0.9914

$$\sigma_n = Gau(200, 30), Washington\ DC\ Mall$$

Thanks！Any Questions?