DSANet: A Deep Supervision-Based Simple Attention Network for Efficient Semantic Segmentation in Remote Sensing Imagery

Wenxu Shi

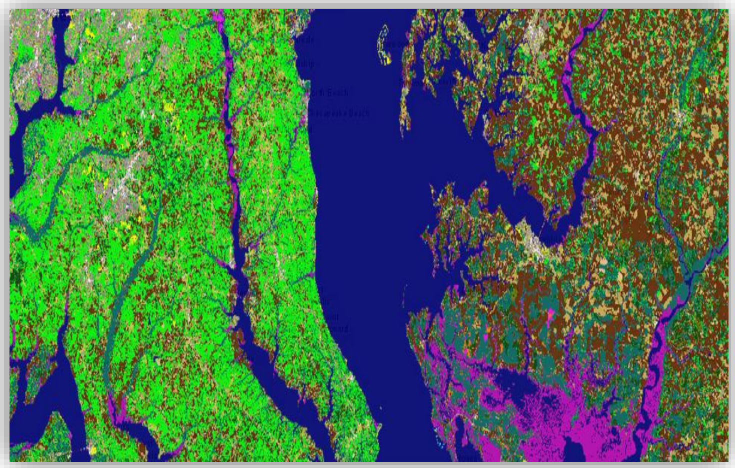
Aerospace Information Research Institute,

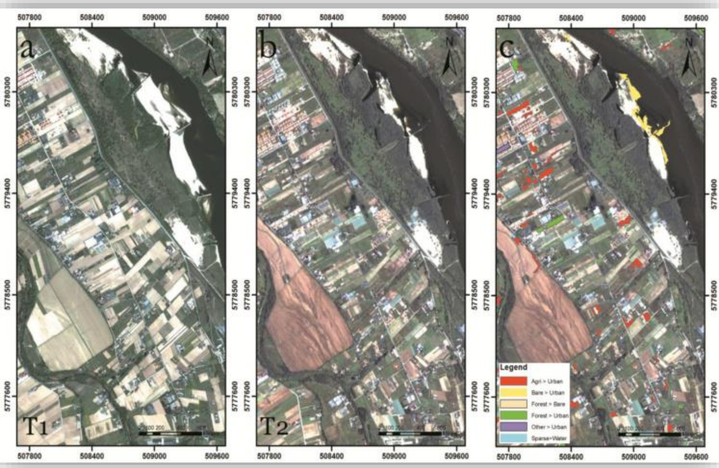
Chinese Academy of Sciences

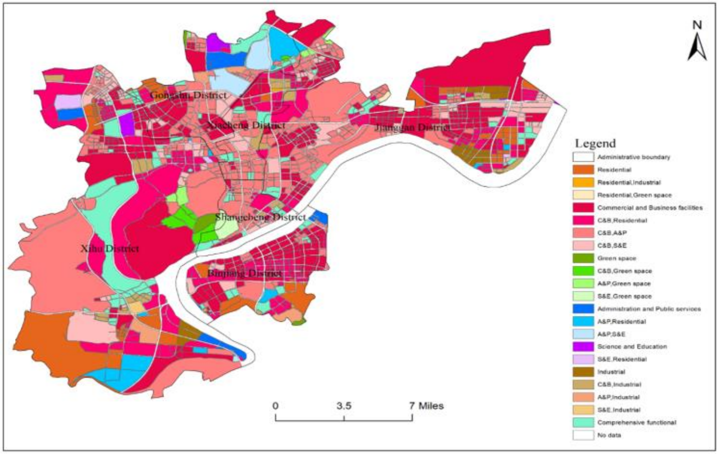
6 Jun. 2023

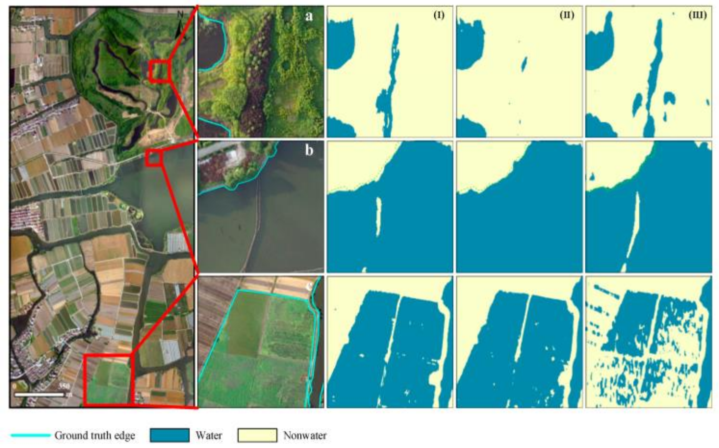
Research Background

Semantic segmentation is a critical task in computer vision, and its special application to remote sensing is RSI interpretation, such as integrated land use and land cover mapping, town change detection, urban functional areas, building footprints, impervious surfaces, and water body extraction.









Research Background

Efficient Semantic Segmentation

“Efficient semantic segmentation can improve the processing efficiency of VHR images”

Researcher	List of research contents
Paszke, Adam, et al.（2016）Enet	Propose a novel deep neural network architecture named ENet (efficient neural network), created specifically for tasks requiring low latency operation.
Li, Gen, et al.（2019）Dabnet	Propose a novel Depth-wise Asymmetric Bottleneck (DAB) module to address this dilemma, which efficiently adopts depth-wise asymmetric convolution and dilated convolution to build a bottleneck structure.
Lo, Shao-Yuan, et al. （2019） EDANet	Propose a novel convolutional network named Efficient Dense modules with Asymmetric convolution (EDANet), which employs an asymmetric convolution structure and incorporates dilated convolution and dense connectivity to achieve high efficiency at low computational cost and model size.
Romera, Eduardo, et al. （2017） Eernet	Propose a deep architecture that is able to run in real time while providing accurate semantic segmentation. The core of our architecture is a novel layer that uses residual connections and factorized convolutions in order to remain efficient while retaining remarkable accuracy.
Fan, Mingyuan, et al. （2021） STDC	Propose a novel and efficient structure named Short-Term Dense Concatenate network (STDC network) by removing structure redundancy.

Research Background

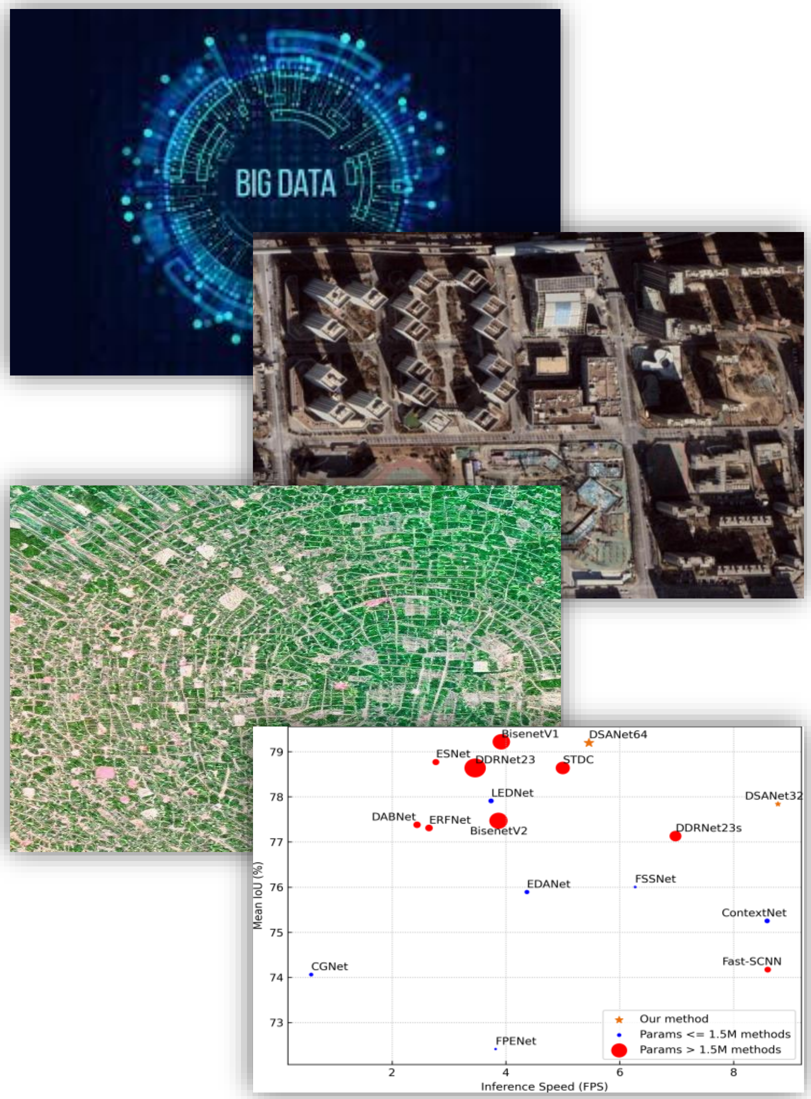
Challenges and Existing problems

（a）**RS Big Data：**

- ➢ High demands on the efficiency of model operation；
- ➢ Hard to deal with very high-resolution (VHR) images;

（b）**Semantic Segmentation：**

- ➢ Lack of details in information modeling ;
- ➢ Inefficient processing;
- ➢ Difficulties for balancing model complexity (inference speed) and segmentation accuracy;

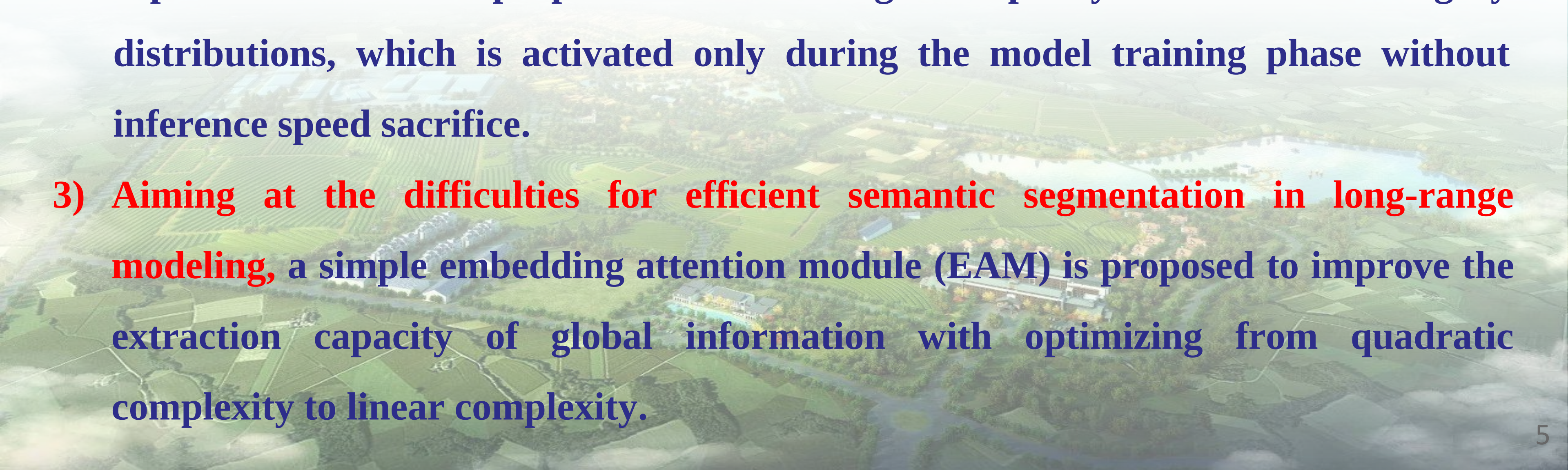


It is urgent to establish a efficient semantic segmentation model with high inference speed and accuracy for VHR images.

Research Background

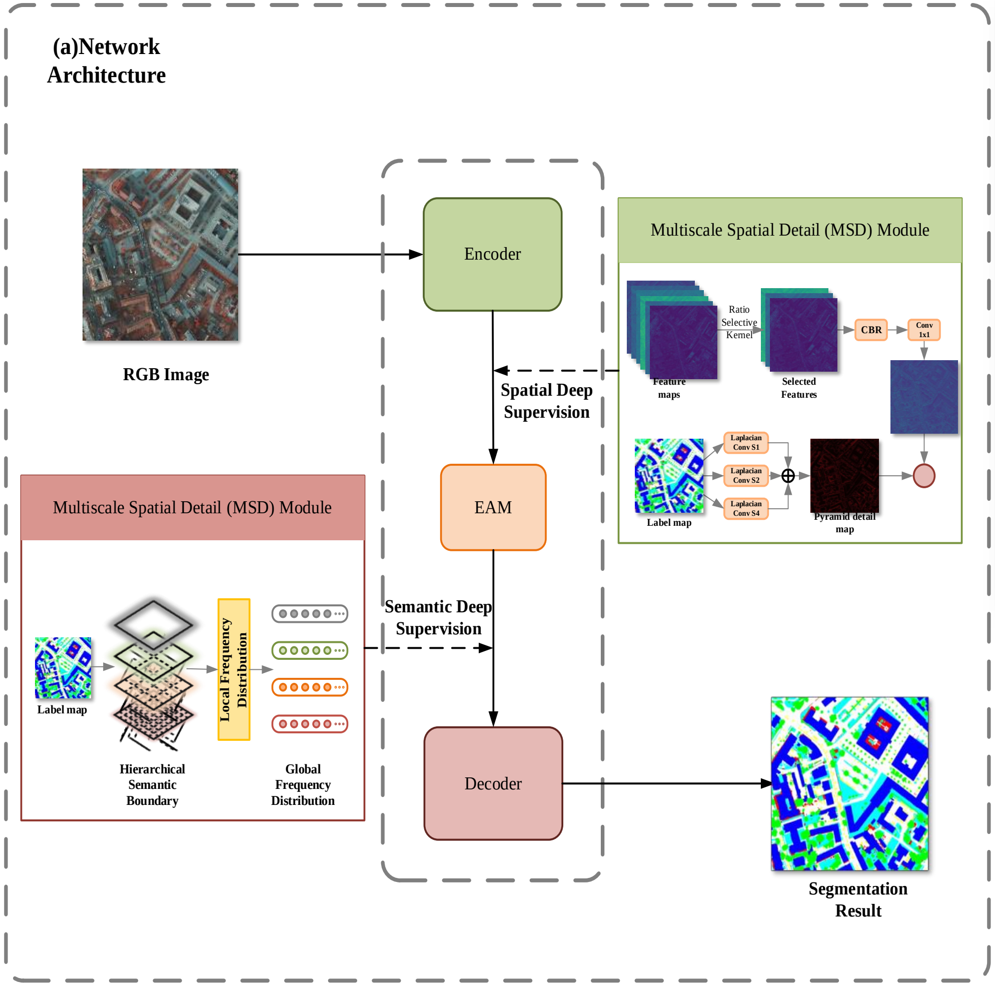
Research objectives

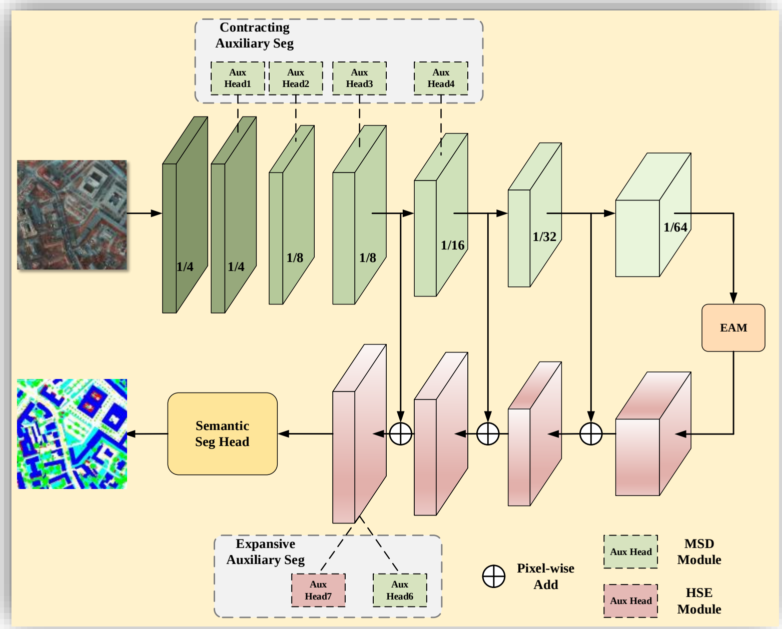
1) Aiming at the lack of spatial details in efficient semantic segmentation information modeling on VHR, the improved multiscale spatial detail (MSD) deep supervision module is proposed to extract rich detail and texture information, which is activated only during the model training phase without inference speed sacrifice.
2) Aiming at the lack of semantic details in efficient semantic segmentation information modeling on VHR, the hierarchical semantic enhancement (HSE) deep supervision module is proposed for enhancing the capacity to discern the category distributions, which is activated only during the model training phase without inference speed sacrifice.
3) Aiming at the difficulties for efficient semantic segmentation in long-range modeling, a simple embedding attention module (EAM) is proposed to improve the extraction capacity of global information with optimizing from quadratic complexity to linear complexity.



Research Content and Technical Route

Framework





DSANet:

1. CNN Lightweight Backbone
2. Embedding Attention (EAM) Module
3. Deep supervision Module MSD and HSE

Research Content and Technical Route

Framework

- Extraction of multilevel convolutional features by a designed low channel capacity, fast downsampling CNN network;
- Feature recalibration of multi-level convolutional features using simple embedding attention module (EAM);
- Spatial detail enhancement of multi-scale convolutional features using a improved multiscale detail enhancement module (MSD) with loss function based on selective kernel;
- Semantic detail enhancement of multiscale convolutional features using a hierarchical semantic enhancement module (HSE) with loss function based on semantic frequency distribution;
- Semantic segmentation of the enhanced features based on the classifier.

Research Content and Technical Route

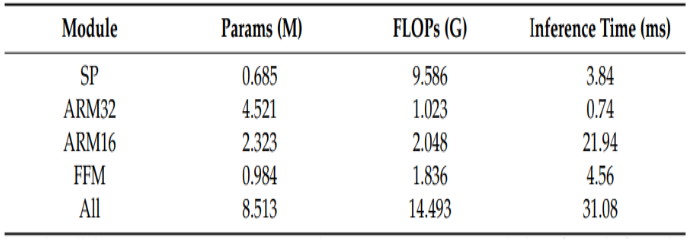
DSANet Backbone

DSANet is an asymmetric, U-shaped, single branch network with an encoder for the contracting path and a decoder for the expansion path.

Observing the inference time spent by a typical two branch network BiSeNet reveals:

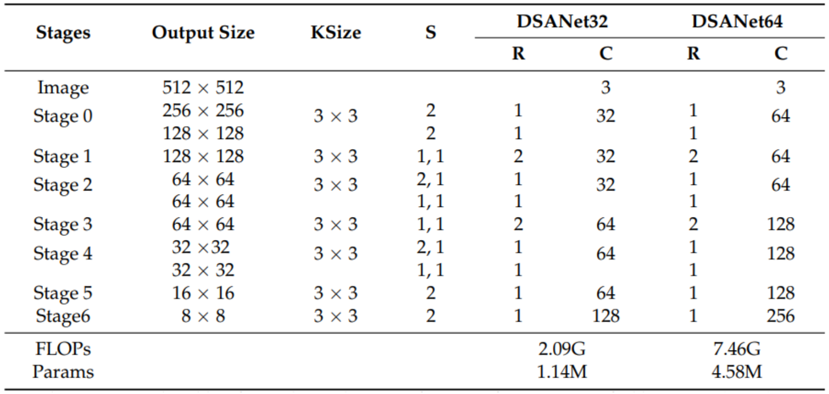
(1) the spatial path (SP) for extracting spatial information, the attention refinement module (ARM) for refining semantic features, and the feature fusion module (FFM) for feature interaction account for more than 30% of the model inference speed;

(2) performing feature operations at the second-to-last scale (ARM16) is extremely time-consuming and unsatisfactory



Module	Params (M)	FLOPs (G)	Inference Time (ms)
SP	0.685	9.586	3.84
ARM32	4.521	1.023	0.74
ARM16	2.323	2.048	21.94
FFM	0.984	1.836	4.56
All	8.513	14.493	31.08

- ✓ **Faster and deeper downsampling;**
- ✓ **Reducing the channel capacity of deeper layers**



Stages	Output Size	KSize	S	DSANet32		DSANet64	
				R	C	R	C
Image	512 × 512				3		3
Stage 0	256 × 256	3 × 3	2	1	32	1	64
	128 × 128		2	1		1	
Stage 1	128 × 128	3 × 3	1, 1	2	32	2	64
Stage 2	64 × 64	3 × 3	2, 1	1	32	1	64
	64 × 64		1, 1	1		1	
Stage 3	64 × 64	3 × 3	1, 1	2	64	2	128
Stage 4	32 ×32	3 × 3	2, 1	1	64	1	128
	32 × 32		1, 1	1		1	
Stage 5	16 × 16	3 × 3	2	1	64	1	128
Stage6	8 × 8	3 × 3	2	1	128	1	256
FLOPs				2.09G		7.46G	
Params				1.14M		4.58M	

Backbone Parameters

Research Content and Technical Route

Embedding Attention Module (EAM)

First, given feature map $\boldsymbol{F} \in R^{C \times H \times W}$;

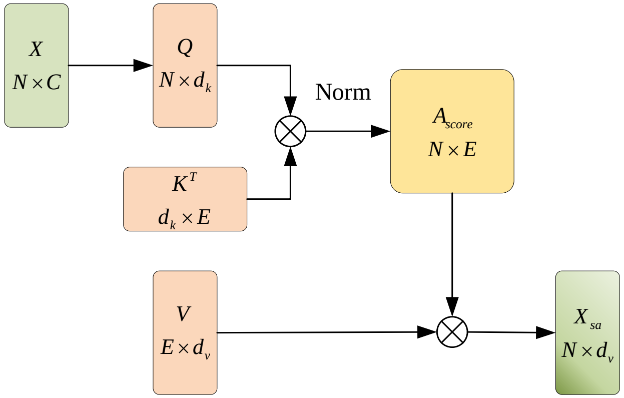
Reshape $\boldsymbol{F}$ to a sequence $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$, where $\boldsymbol{x}_i \in R^C$ is the feature vector of element N;

Perform linear transformations on $\boldsymbol{X}$ to obtain query matrix $\boldsymbol{Q} \in R^{N \times d_k}$:

$$\boldsymbol{Q} = W_Q(\boldsymbol{X})$$

Memoried key matrix $\boldsymbol{K} \in R^{N \times d_k}$, and value matrix $\boldsymbol{V} \in R^{N \times d_v}$ are pre-generated, where $d_v = d_k$, and are retained until subsequent calculations.

Calculate the cosine similarity between the i-th element and the j-th element as $(q_i^T k_j)$. The attention score $\widehat{\boldsymbol{A}}_{i,j}$ of matrix $\boldsymbol{Q}$ and $\boldsymbol{K}$ is defined as:



$$\boldsymbol{A}_{i,j} = \sum_t^{d_k} q_{i,t} \cdot k_{t,j}$$

$$\widetilde{\boldsymbol{A}}_{i,j} = softmax(\boldsymbol{Q}, \boldsymbol{K})_{i,j} = \frac{\exp(\boldsymbol{A}_{i,j})}{\sum_k^E \exp(\boldsymbol{A}_{k,j})}$$

L1 normalization is specifically applied following softmax activation.

$$\widehat{\boldsymbol{A}}_{i,j} = Norm_{L1}(\widetilde{\boldsymbol{A}}_{i,j}) = \frac{\widetilde{\boldsymbol{A}}_{i,j}}{\sum_k^{d_k} \widetilde{\boldsymbol{A}}_{i,k}}$$

Obtain $\boldsymbol{X}_{sa}$ by multiplying $\boldsymbol{V}$ with $\widehat{\boldsymbol{A}}$:

$$\boldsymbol{X}_{sa} = \widehat{\boldsymbol{A}}\boldsymbol{V}$$

Research Content and Technical Route

Multiscale Detail Enhancement (MSD)

First, given feature map $\boldsymbol{F_{in}} \in R^{C\times H\times W}$ of shallow layer;

Obtain selected feature maps $\boldsymbol{F_S} \in R^{rC\times H\times W}$ through a selective kernel, where $\boldsymbol{r}$ is selective ratio;

Obtain $\boldsymbol{F_S} \in R^{H\times W}$ with channel dimension 1 by a 3×3 convolution and a 1×1 convolution;

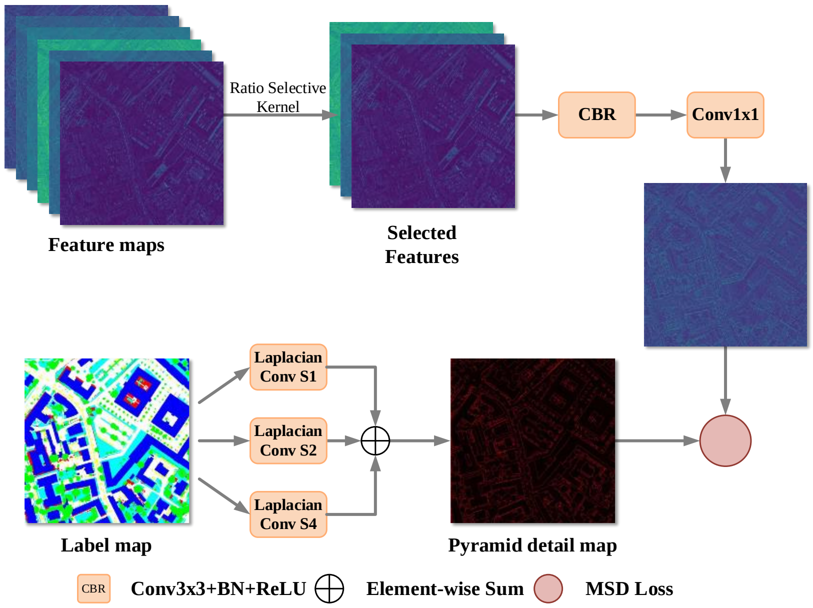
Set the discrete Laplace operator $\boldsymbol{O}$ as edge extractor:

$$\boldsymbol{O} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Create multiscale detail maps $\boldsymbol{D_0} \in R^{H\times W}$, $\boldsymbol{D_2} \in R^{H\times W}$, and $\boldsymbol{D_4} \in R^{H\times W}$ performed by Laplace convolution operators with varying strides from the label map;

Obtain pyramid detail map $\boldsymbol{P} \in R^{H\times W}$ by summing multiscale details maps:

$$P = D_0 + D_2 + D_4$$



Use binary cross-entropy (BCE) loss with category proportion-insensitive Dice loss to evaluate the similarity of selected feature maps $\boldsymbol{F_S}$ and pyramid detail map $\boldsymbol{P}$.

Research Content and Technical Route

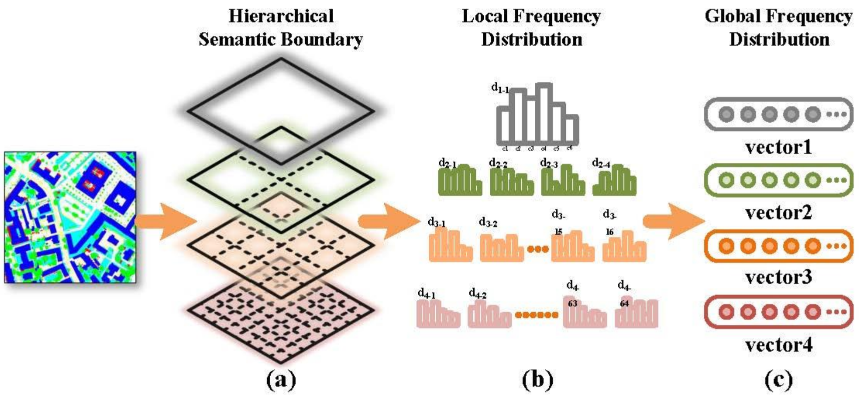
Hierarchical Semantic Enhancement (HSE)

First, given label map $y \in R^{H \times W}$;

1. Set hierarchical semantic boundary. Assume that $\boldsymbol{N}$ boundary levels and the boundary level of $\boldsymbol{n}$ slice the label map in $2^{\boldsymbol{n}}$ patches along the length and width, respectively. The label patches are set as $y_{\boldsymbol{n}} \in \{y_1, y_2, \dots, y_N\}$, where $y_{\boldsymbol{n}} = \{y_{\boldsymbol{n}}^{(j)}\}_{j=1}^{2^{2\boldsymbol{n}}}$ is the set of label patches in level $\boldsymbol{n}$ and $y_{\boldsymbol{n}}^{(j)} \in R^{\frac{H}{2^n} \times \frac{W}{2^n}}$;

2. Calculate the local frequency distribution. The category distribution $d_{\boldsymbol{n}}^{(j)}$ is calculated separately for each label patch $y_{\boldsymbol{n}}^{(j)}$, where j is the sequence number of the label patch set;

3. Aggregate the global distribution vector. The label patches at boundary level $\boldsymbol{n}$ are concatenated to generate the global frequency distribution vector $\hat{v}_{\boldsymbol{n}}$. The HSE vector of label map $\hat{v} = \{\hat{v}_{\boldsymbol{n}}\}_{\boldsymbol{n}=0}^{\boldsymbol{N}}$;



4. Repeat step 1-3 for feature map $\boldsymbol{F}$ to obtain global frequency distribution vector $v_{\boldsymbol{n}}$ and its HSE vector v;

5. Use binary cross-entropy (BCE) loss to evaluate the similarity of $\hat{v}$ and v.

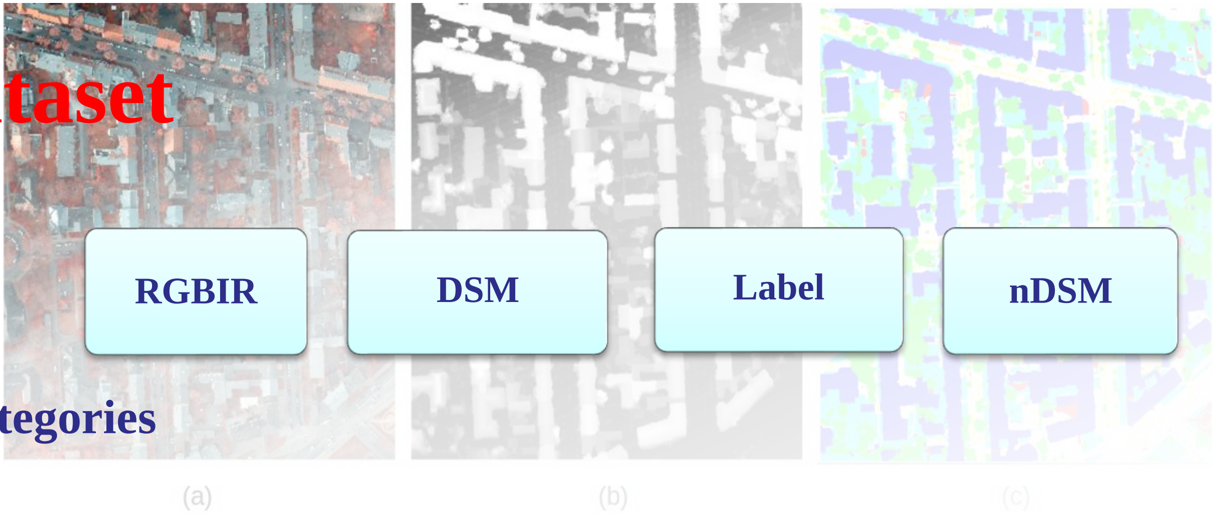
Research Content and Technical Route

- ## Potsdam Dataset

ISPRS Website

5 cm spatial resolution

Urban, 38 Patches, 6 Categories



RGBIR

DSM

Label

nDSM

- ## Vaihingen Dataset

ISPRS Website

9 cm spatial resolution

Village, 33 Patches, 6 Categories

RGBIR

DSM

Label

nDSM

Experiment Results

Experiment Settings

Experimental Parameter Setting

- SGD is chosen as the optimizer, the total number of training times $iter_0$ is 80,000, the initial learning rate lr_0 is 0.001, and the learning rate lr is updated using the "poly" training strategy, $lr = lr_0\left(1-\frac{iter}{iter_0}\right)^{power}$, where $power$ is set to 0.9, the batch size is 16.

Data Preprocessing and Augmentation

Data Preprocessing

- **Data Cropping**： Crop the raw images to a size of 500×500 pixels with a stride equal to half the size of the cropped image；
- Discard images with lengths or widths that are less than a quarter of the cropped image size；

Data Augmentation

- **Multiscale Resizing:** A scale number is randomly selected from 0.5, 0.75, 1.0, 1.25, 1.5, 1.75 and 2.0 for the height and width resizing.；
- **Random Cropping:** Randomly select a 512×512 pixel block of data and crop it；
- **Random Flipping:** Includes vertical flip and horizontal flip；
- **Photometric Distortion:** Randomly adjust the brightness, contrast, saturation and hue levels of the images；
- **Normalization:** Adjust the data distribution to conform to a normal distribution。

Experiment Results

Ablation Experiment

To quantify the role of the EAM, MSD and HSE modules, ablation experiments were conducted on the ISPRS Potsdam dataset.

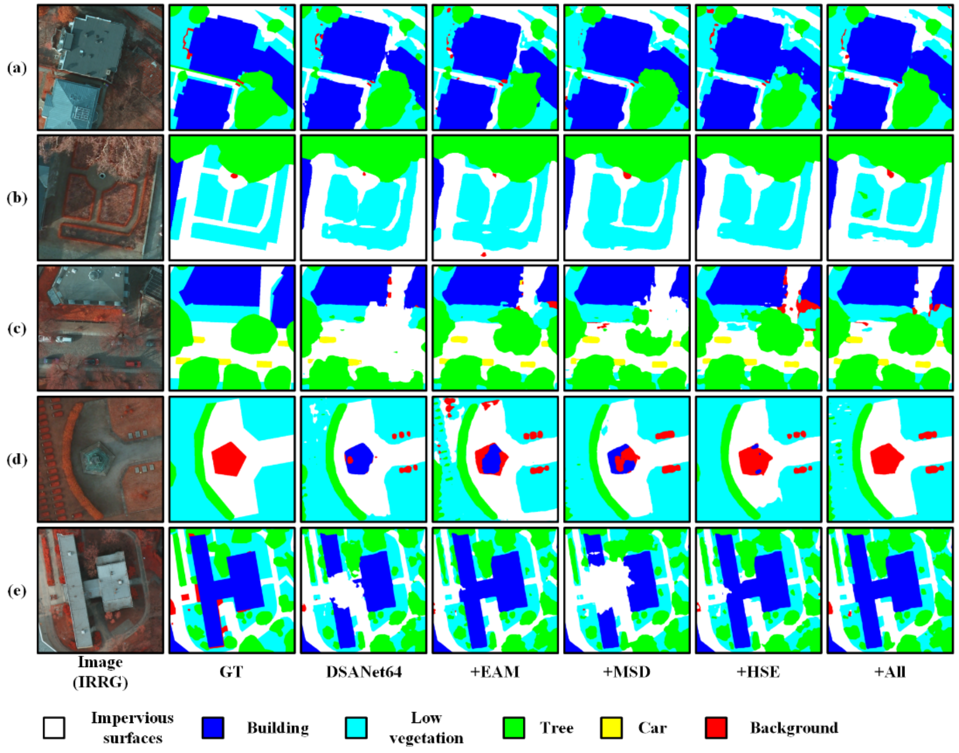
Method	EAM	MSD	HSE	mIoU (%)	mF1 (%)
DSANet64				77.05	86.90
	√			78.17	87.60
		√		77.96	87.48
			√	77.33	87.39
	√	√		79.06	88.17
	√	√	√	79.20	88.25

DSANet64 obtained 78.17%, 77.96 % and 77.33% of mIoUs using EAM, MSD and HSE modules, respectively, which are 1.12%, 0.91% and 0.28% higher compared to DSANet64 backbone network, demonstrating the effectiveness of DSANet.

Experiment Results

Ablation Experiment

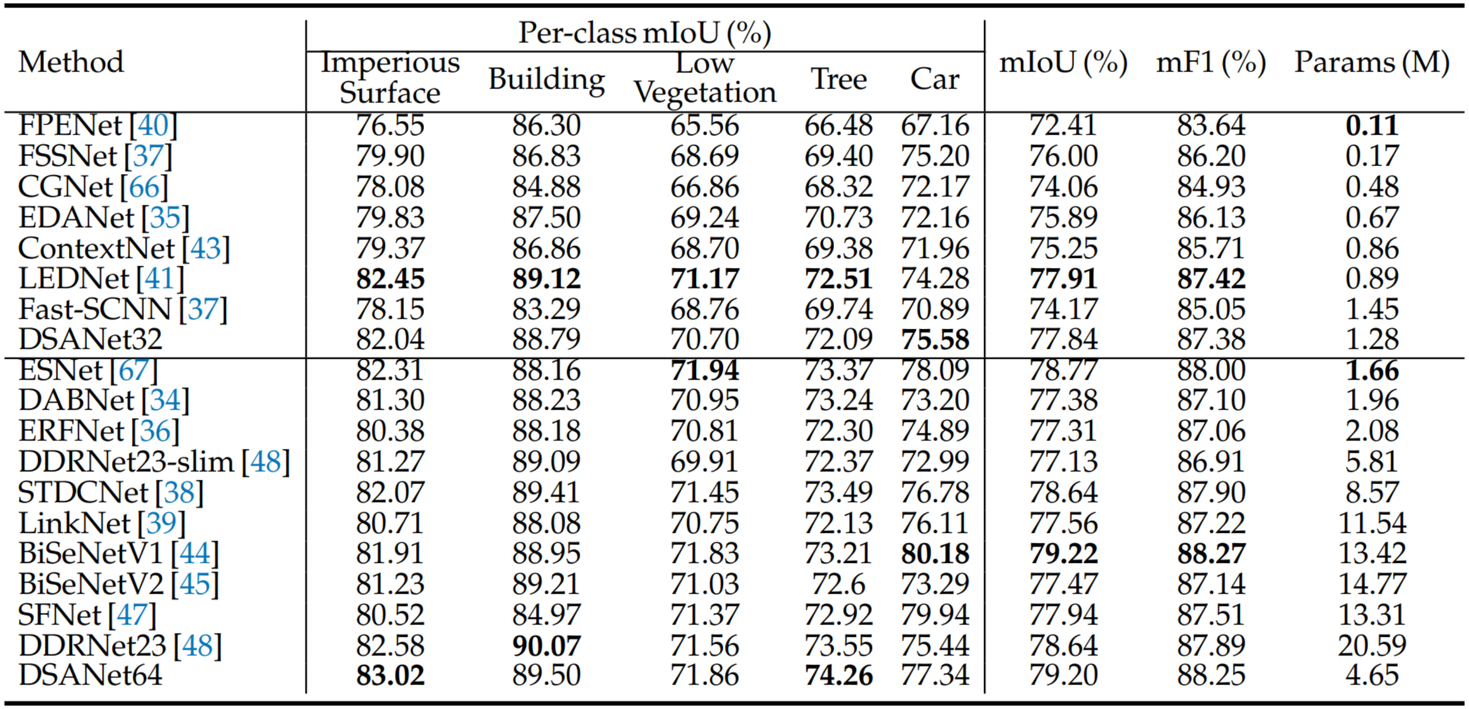
In order to qualitatively understand the role of the EAM, MSD and HSE modules through visual representation, ablation experiments were conducted on multiple different features selected from the ISPRS Potsdam dataset.



DSANet64 can obtain better segmentation results than the backbone network using EAM, MSD and HSE modules, respectively, and on balance, DSANet64 is effective.

Experiment Results

Comparison-Potsdam



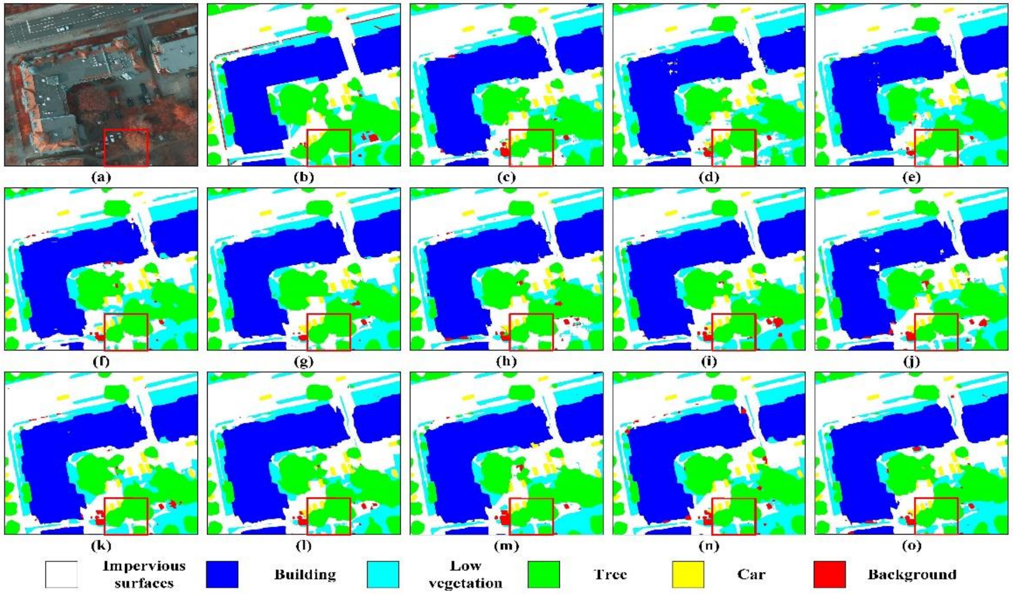
Method	Per-class mIoU (%)					mIoU (%)	mF1 (%)	Params (M)
	Imperious Surface	Building	Low Vegetation	Tree	Car			
FPENet [40]	76.55	86.30	65.56	66.48	67.16	72.41	83.64	**0.11**
FSSNet [37]	79.90	86.83	68.69	69.40	75.20	76.00	86.20	0.17
CGNet [66]	78.08	84.88	66.86	68.32	72.17	74.06	84.93	0.48
EDANet [35]	79.83	87.50	69.24	70.73	72.16	75.89	86.13	0.67
ContextNet [43]	79.37	86.86	68.70	69.38	71.96	75.25	85.71	0.86
LEDNet [41]	**82.45**	**89.12**	**71.17**	**72.51**	74.28	**77.91**	**87.42**	0.89
Fast-SCNN [37]	78.15	83.29	68.76	69.74	70.89	74.17	85.05	1.45
DSANet32	82.04	88.79	70.70	72.09	**75.58**	77.84	87.38	1.28
ESNet [67]	82.31	88.16	**71.94**	73.37	78.09	78.77	88.00	**1.66**
DABNet [34]	81.30	88.23	70.95	73.24	73.20	77.38	87.10	1.96
ERFNet [36]	80.38	88.18	70.81	72.30	74.89	77.31	87.06	2.08
DDRNet23-slim [48]	81.27	89.09	69.91	72.37	72.99	77.13	86.91	5.81
STDCNet [38]	82.07	89.41	71.45	73.49	76.78	78.64	87.90	8.57
LinkNet [39]	80.71	88.08	70.75	72.13	76.11	77.56	87.22	11.54
BiSeNetV1 [44]	81.91	88.95	71.83	73.21	**80.18**	**79.22**	**88.27**	13.42
BiSeNetV2 [45]	81.23	89.21	71.03	72.6	73.29	77.47	87.14	14.77
SFNet [47]	80.52	84.97	71.37	72.92	79.94	77.94	87.51	13.31
DDRNet23 [48]	82.58	**90.07**	71.56	73.55	75.44	78.64	87.89	20.59
DSANet64	**83.02**	89.50	71.86	**74.26**	77.34	79.20	88.25	4.65

- **DSANet32 and DSANet64 achieve better and suboptimal results for both small and large model comparisons, with mIoU of 77.84% and 79.20%, respectively.**
- **The segmentation results of most other efficient segmentation networks are significantly lower than DSANet, which also validates the effectiveness of DSANet.**

Experiment Results

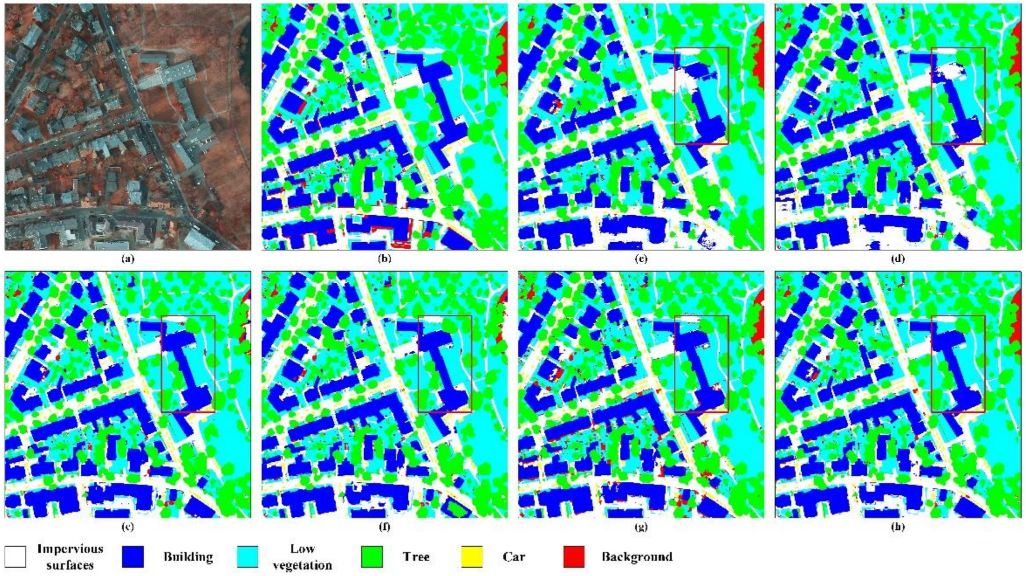
Comparison-Potsdam

(a) IRRG image, (b) GT, (c) FPENet, (d) FSSNet, (e) CGNet, (f) ContextNet, (g) Fast-SCNN, (h) ERFNet, (i) STDC1, (j) LinkNet, (k) ICNet34, (l) BiSeNet V1, (m) SFNet, (n)DDRNet23, (o) DSANet64.



Potsdam-small

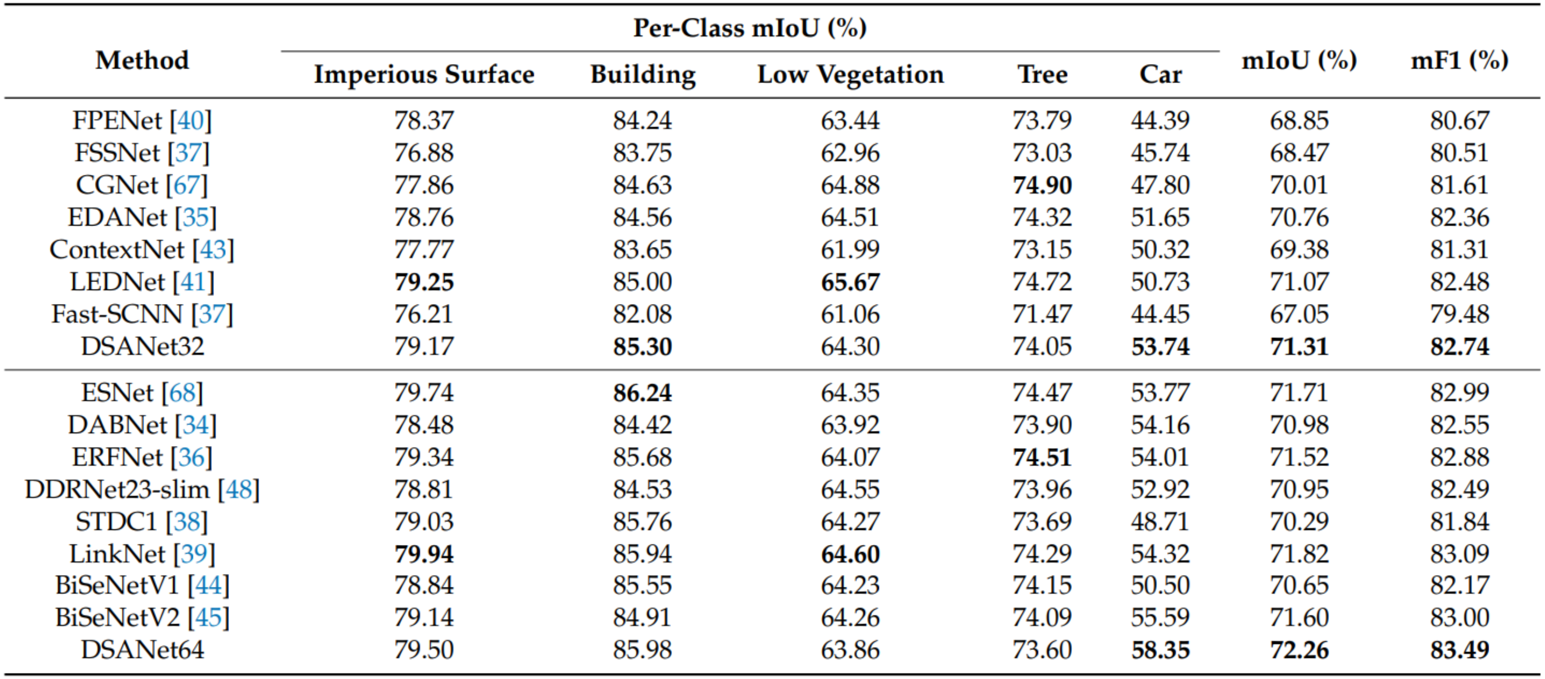
Potsdam-large



(a) IRRG image, (b) GT, (c) FPENet, (d) ERFNet, (e)DDRNet23-slim, (f) STDC1, (g) BiSeNet V1, (h) BiSeNet V2, (i) DSANet64.

Experiment Results

Comparison-Vaihingen



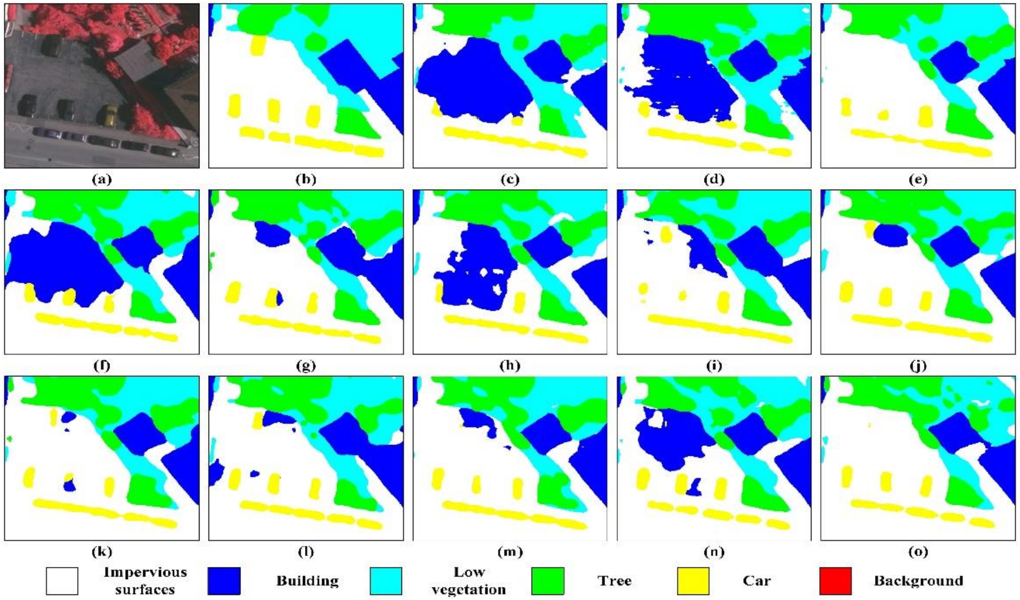
Method	Per-Class mIoU (%)					mIoU (%)	mF1 (%)
	Imperious Surface	**Building**	**Low Vegetation**	**Tree**	**Car**		
FPENet [40]	78.37	84.24	63.44	73.79	44.39	68.85	80.67
FSSNet [37]	76.88	83.75	62.96	73.03	45.74	68.47	80.51
CGNet [67]	77.86	84.63	64.88	**74.90**	47.80	70.01	81.61
EDANet [35]	78.76	84.56	64.51	74.32	51.65	70.76	82.36
ContextNet [43]	77.77	83.65	61.99	73.15	50.32	69.38	81.31
LEDNet [41]	**79.25**	85.00	**65.67**	74.72	50.73	71.07	82.48
Fast-SCNN [37]	76.21	82.08	61.06	71.47	44.45	67.05	79.48
DSANet32	79.17	**85.30**	64.30	74.05	**53.74**	**71.31**	**82.74**
ESNet [68]	79.74	**86.24**	64.35	74.47	53.77	71.71	82.99
DABNet [34]	78.48	84.42	63.92	73.90	54.16	70.98	82.55
ERFNet [36]	79.34	85.68	64.07	**74.51**	54.01	71.52	82.88
DDRNet23-slim [48]	78.81	84.53	64.55	73.96	52.92	70.95	82.49
STDC1 [38]	79.03	85.76	64.27	73.69	48.71	70.29	81.84
LinkNet [39]	**79.94**	85.94	**64.60**	74.29	54.32	71.82	83.09
BiSeNetV1 [44]	78.84	85.55	64.23	74.15	50.50	70.65	82.17
BiSeNetV2 [45]	79.14	84.91	64.26	74.09	55.59	71.60	83.00
DSANet64	79.50	85.98	63.86	73.60	**58.35**	**72.26**	**83.49**

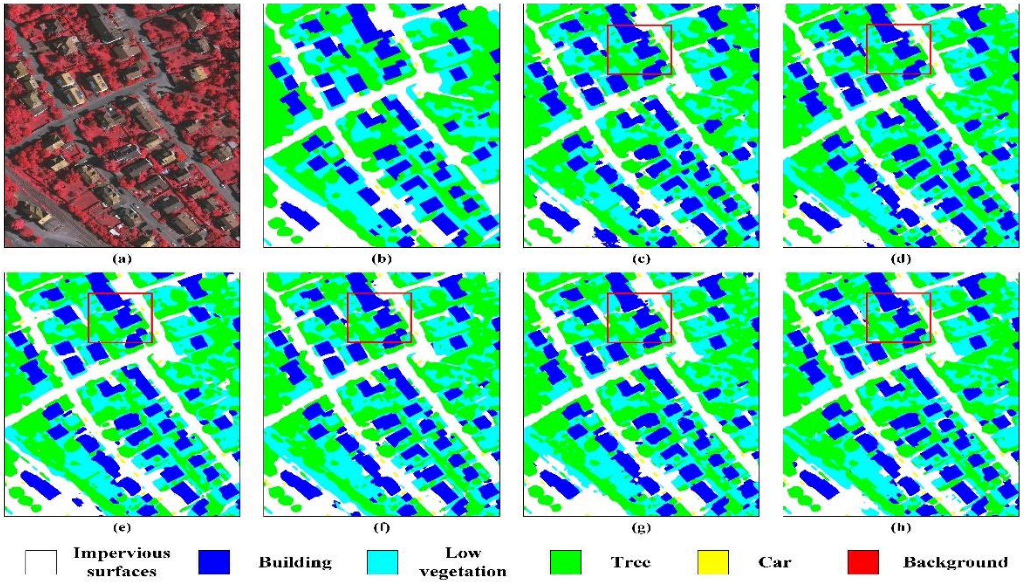
- **DSANet32 and DSANet64 achieve optimal results for both small and large model comparisons, with mIoU of 71.31% and 72.26%, respectively.**
- **The segmentation results of other efficient segmentation networks are significantly lower than DSANet, which again validates the effectiveness of DSANet.**

Experiment Results

Comparison-Vaihingen

(a) IRRG image, (b) GT, (c) FPENet, (d) FSSNet, (e) CGNet, (f) ContextNet, (g) Fast-SCNN, (h) ESNet, (i) ERFNet, (j) DDRNet23-slim, (k) STDC1, (l)LinkNet, (m) BiSeNet V1, (n) BiSeNet V2, (o) DSANet64.

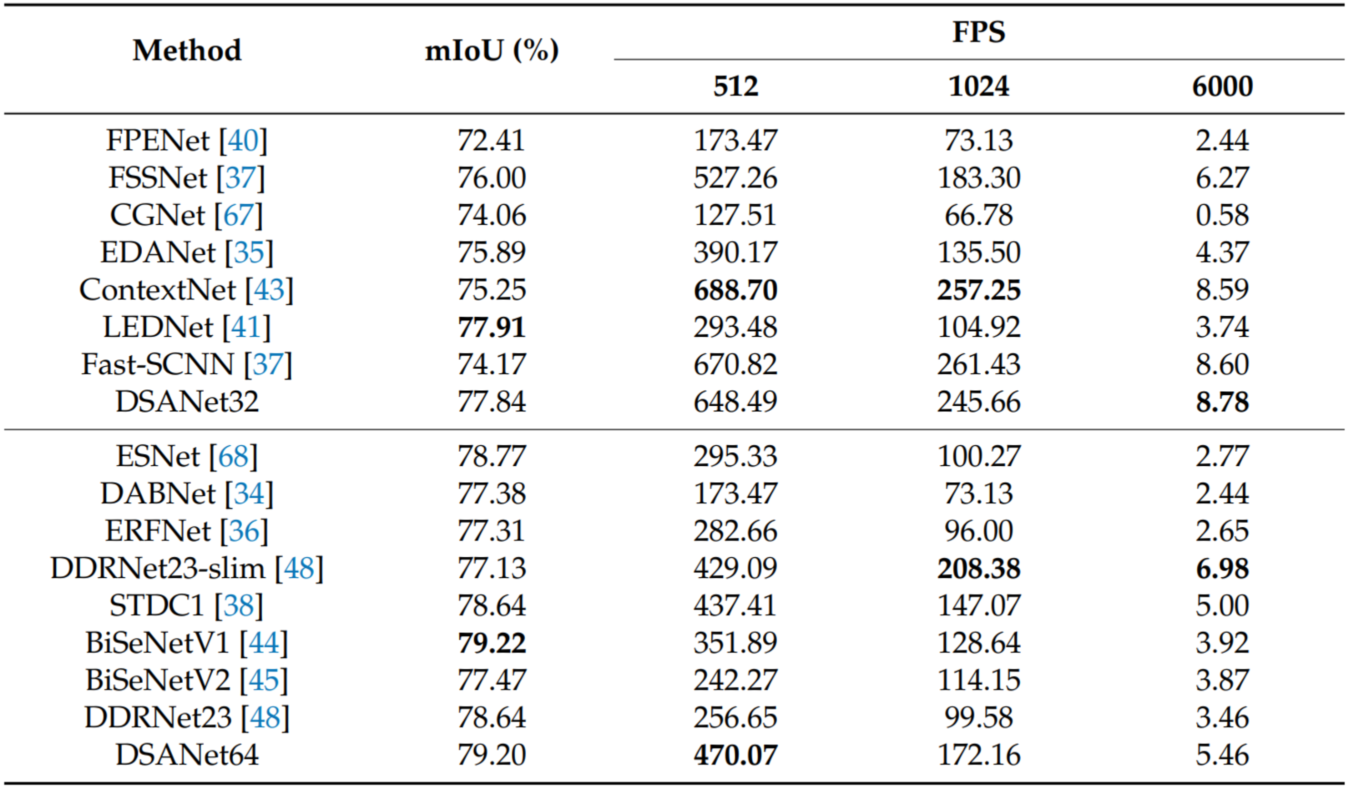




(a) IRRG image, (b) GT, (c) FPENet, (d) ERFNet, (e)DDRNet23-slim, (f) STDC1, (g) BiSeNet V1, (h) BiSeNet V2, (i) DSANet64.

Experiment Results

Inference Speed



Method	mIoU (%)	FPS		
		512	1024	6000
FPENet [40]	72.41	173.47	73.13	2.44
FSSNet [37]	76.00	527.26	183.30	6.27
CGNet [67]	74.06	127.51	66.78	0.58
EDANet [35]	75.89	390.17	135.50	4.37
ContextNet [43]	75.25	**688.70**	**257.25**	8.59
LEDNet [41]	**77.91**	293.48	104.92	3.74
Fast-SCNN [37]	74.17	670.82	261.43	8.60
DSANet32	77.84	648.49	245.66	**8.78**
ESNet [68]	78.77	295.33	100.27	2.77
DABNet [34]	77.38	173.47	73.13	2.44
ERFNet [36]	77.31	282.66	96.00	2.65
DDRNet23-slim [48]	77.13	429.09	**208.38**	**6.98**
STDC1 [38]	78.64	437.41	147.07	5.00
BiSeNetV1 [44]	**79.22**	351.89	128.64	3.92
BiSeNetV2 [45]	77.47	242.27	114.15	3.87
DDRNet23 [48]	78.64	256.65	99.58	3.46
DSANet64	79.20	**470.07**	172.16	5.46

- **DSANet32 achieves the optimal inference speed of 8.78 FPS on large images and DSANet64 achieves the optimal inference speed of 470.07 FPS on small images, in addition, DSANet has better inference speed on data of different scales.**

Experiment Results

Conclusion

- Lightweight DSANet is proposed for semantic segmentation of remote sensing images. DSANet better balances the contradiction between the operation efficiency and accuracy；
- Using multiscale spatial detail enhancement and hierarchical semantic enhancement modules to effectively enhance the model's ability to extract detailed and semantic information without sacrificing inference speed；
- A simple embedding attention module (EAM) with linear complexity performs long-range relationship modeling；
- DSANet still has room for optimization in terms of operational efficiency and accuracy, such as the use of knowledge distillation, structural re-parameterization and model pruning.

PanDiff: A Novel Pansharpening Method Based on Denoising Diffusion Probabilistic Model

Wenxu Shi

Aerospace Information Research Institute,

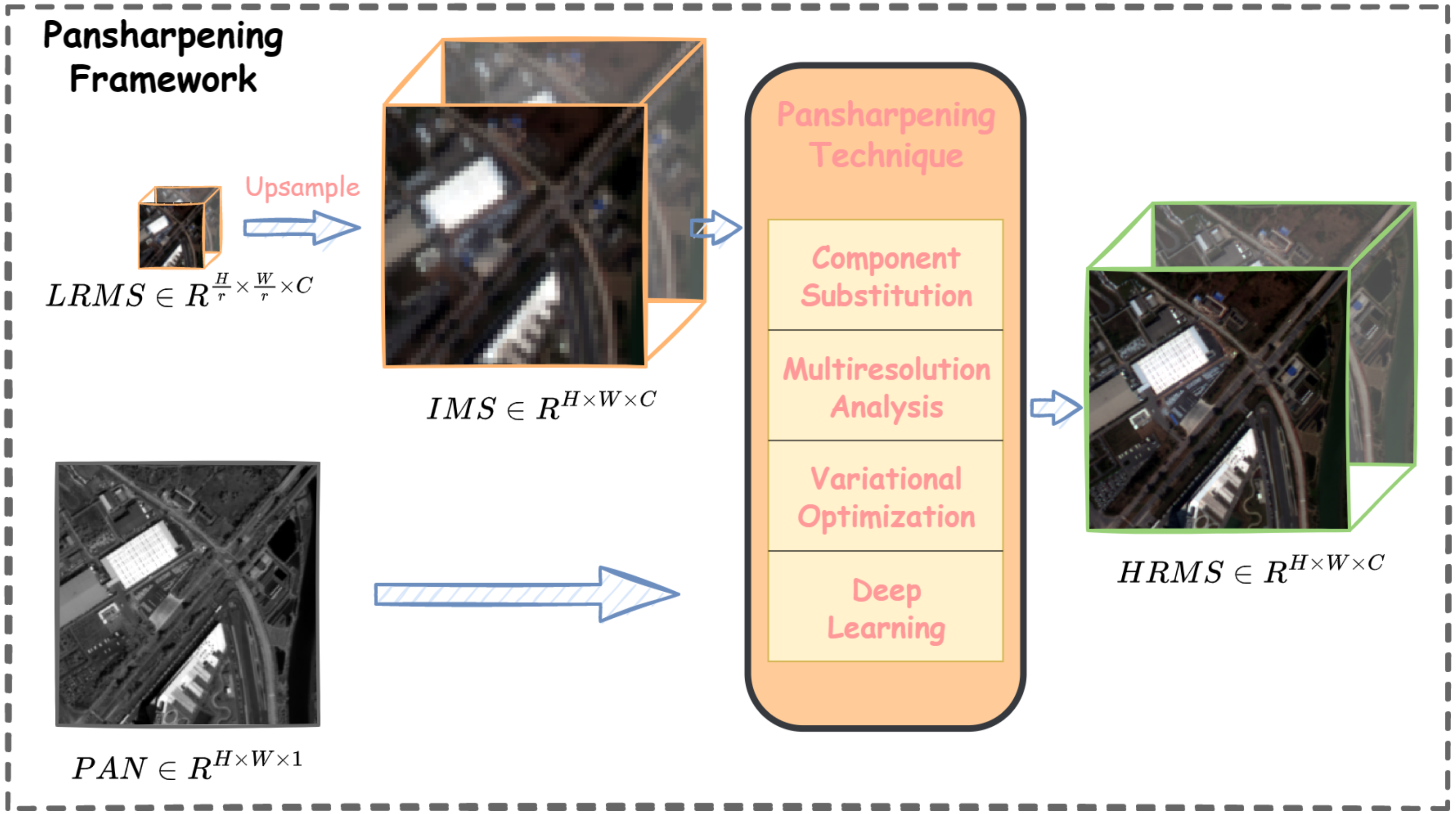
Chinese Academy of Sciences

6 Jun. 2023

Research Background

Pansharpening is a technique used in remote sensing and image processing to obtain the high-spatial-resolution (HR) multispectral (MS) images by fusing HR panchromatic (PAN) images and lower-spatial-resolution (LR) MS images.

$$HRMS = F_\theta(LRMS, PAN)$$


Pansharpening Framework
Upsample
$LRMS \in R^{\frac{H}{r} \times \frac{W}{r} \times C}$
$IMS \in R^{H \times W \times C}$
$PAN \in R^{H \times W \times 1}$
Pansharpening Technique
Component Substitution
Multiresolution Analysis
Variational Optimization
Deep Learning
$HRMS \in R^{H \times W \times C}$

Research Background

Traditional Pansharpening

Researcher	List of research contents
Laben, and Bernard.（2000） **CS**. Gram-Schmidt	Perform a ***Gram-Schmidt transformation*** on the simulated lower spatial resolution panchromatic image and the plurality of lower spatial resolution spectral band images.
Shahdoosti, and Hassan.（2016） **CS**. PCA	Propose a new consistent data transformation method in spatial domain, this paper applies the ***PCA transform*** to the spatial information of the neighboring pixels.
Otazu, González-Audícana, et al.（2005）**MRA**. AWLP	Present a technique which takes into account the physical electromagnetic spectrum responses of sensors during the fusion process, which produces images closer to the image obtained by the ideal sensor than those obtained by usual ***wavelet-based*** image fusion methods.
Aiazzi, Alparone, et al. （2006） **MRA**. MTF-GLP	A model of the ***modulation transfer functions (MTF)*** of the multispectral scanner is exploited to design the GLP reduction filter.
Ballester, Vicent, et al. （2006） **VO**. P+XS	Based on the assumption that, to a large extent, the geometry of the spectral channels is contained in the ***topographic map*** of its panchromatic image.
Li, and Yang. (2010) **VO**. Sparse Representation	Address the remote sensing image pan-sharpening problem from the perspective of ***compressed sensing theory*** which ensures that with the sparsity regularization, a compressible signal can be correctly recovered from the global linear sampled data.

Research Background

DL-based Pansharpening

Researcher	List of research contents
Masi, Cozzolino, et al.（2016） PNN	A new pansharpening method is proposed, based on ***convolutional neural networks***. We adapt a simple and effective three-layer architecture recently proposed for super-resolution to the pansharpening problem.
Yang, Fu, et al.（2017）PanNet	Propose a deep network architecture for the pan-sharpening problem called PanNet. We incorporate ***domain-specific knowledge*** to design our PanNet architecture by focusing on the two aims of the pan-sharpening problem: spectral and spatial preservation.
Wei, Yuan, et al. （2017） DRPNN	The concept of ***residual learning*** is introduced to form a very deep convolutional neural network to make the full use of the high nonlinearity of the deep learning models.
Liu, Liu, et al. （2020）TFNet	Propose a ***Two-stream*** Fusion Network (TFNet) to address the problem of pan-sharpening. …the proposed TFNet aims to fuse PAN and MS images in feature domain and reconstruct the pan-sharpened image from the fused features.
Meng, Wang, et al. （2022） Vision Transformer	Propose an improved and advanced purely ***transformer-based model*** for pansharpening.
Ma, Yu, et al. (2020) PanGan	Propose a novel ***unsupervised framework*** for pan-sharpening based on a ***generative adversarial network***, termed as Pan-GAN, which does not rely on the so-called ground-truth during network training.

Research Background

Existing Problems

（a） **Component Substitution (CS)-based Method**

- High spatial quality, Low spectral fidelity

（b） **Multiresolution Analysis (MRA)-based Method**

- High spectral fidelity, Low spatial quality

（c） **Variational Optimization (VO)-based Method**

- Computationally costly
- Underlying assumptions not always match the fusion situation

Limited by strong physical assumption!

（d） **Deep Learning (DL)-based Method：**

- CNN: Tend to smooth features
- Transformer: Need large dataset for model training
- GAN: Unstable training

It is urgent to establish a new pansharpening method to avoid above problems.

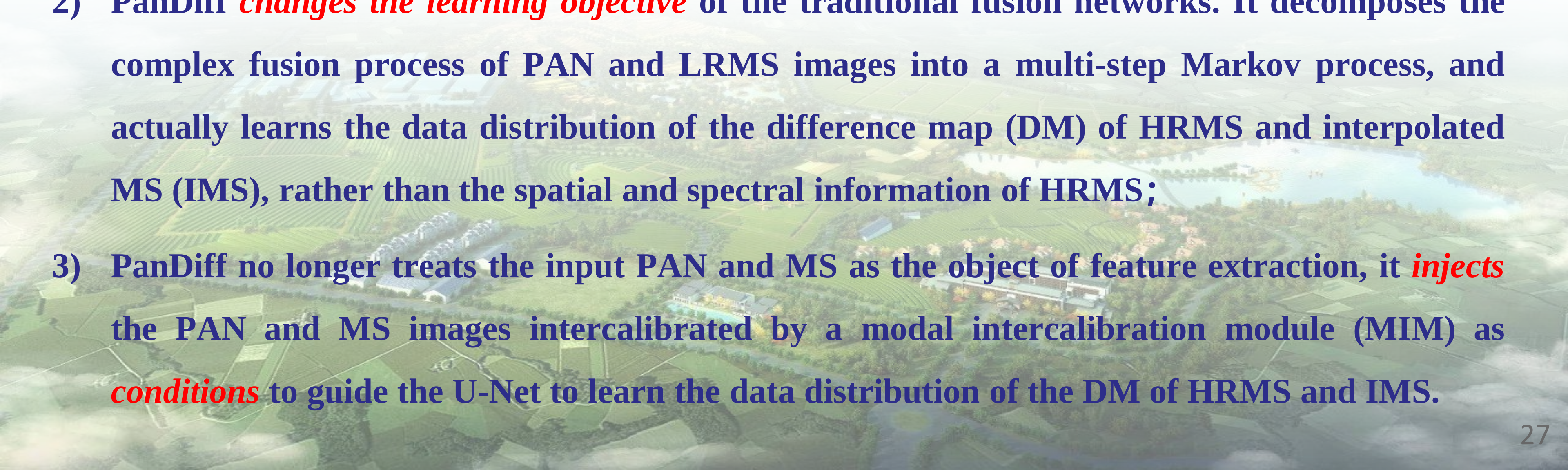
Content and Technical Route

Solution

A novel denoising diffusion probabilistic model (DDPM)-based pansharpening model (PanDiff) is proposed from a fresh perspective to avoid the inherent flaws of the traditional and DL-based approaches.

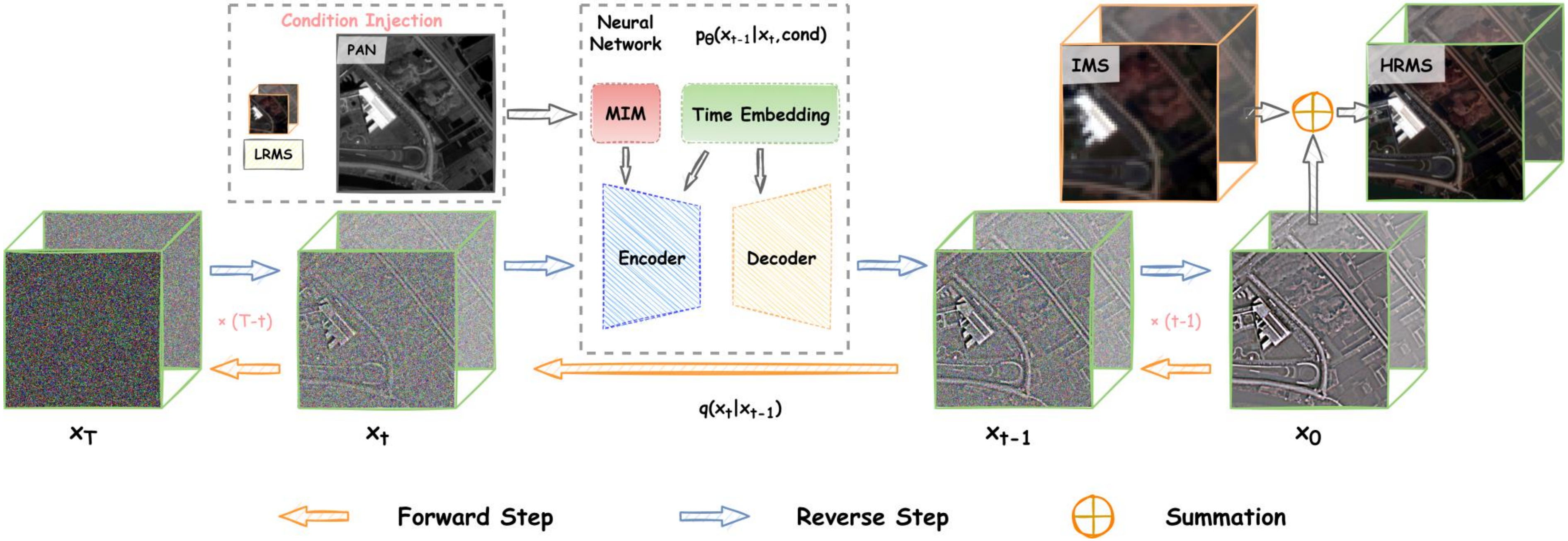
Contributions

1) **PanDiff is a *generative model* based on the DDPM which is *first designed* for pansharpening;**
2) **PanDiff *changes the learning objective* of the traditional fusion networks. It decomposes the complex fusion process of PAN and LRMS images into a multi-step Markov process, and actually learns the data distribution of the difference map (DM) of HRMS and interpolated MS (IMS), rather than the spatial and spectral information of HRMS；**
3) **PanDiff no longer treats the input PAN and MS as the object of feature extraction, it *injects* the PAN and MS images intercalibrated by a modal intercalibration module (MIM) as *conditions* to guide the U-Net to learn the data distribution of the DM of HRMS and IMS.**



Content and Technical Route

Framework



Notations

1. $q(\cdot|\cdot)$: Forward (Diffusion) step

2. $p_\theta(\cdot|\cdot)$: Reverse (Denoised) step with network θ

3. t : Discrete timesteps t on the range of [0, T]

4. x_0 : Prior distribution of data $GT - IMS$

5. x_t : Diffused data (latent state) at step t

6. x_T : Random noise after diffusion

PanDiff:

1. DDPM: Forward Step

2. DDPM: Reverse Step

3. Condition Injection Branch

4. Modal Intercalibration

Content and Technical Route

DDPM: Forward Process

Given prior data distribution $\boldsymbol{q(x_0)}$;

Continuously **adding Gaussian noise** to latent states x_t on Markov chain:

$$q\,(x_t \mid x_{t-1}) = N\left(x_t; \sqrt{1-\beta_t}x_{t-1}, \sqrt{\beta_t}I\right)(1)$$

where β_t represents the variance of the added Gaussian noise in the transition process from x_{t-1} to x_t, and all the variance schedule $\beta_1, \ldots, \beta_T \in [0,1)$;

Obtain an **approximate standard normal distribution $\boldsymbol{x_T}$** $\sim N(0, I)$ after continuously Gaussian transition $q(x_t \mid x_{t-1})$;

The forward diffusion process is given by the approximate posterior:

$$q(x_{1:T} \mid x_0) = \prod_{t=1}^{T} q(x_t \mid x_{t-1})\,(2)$$

The latent state $\boldsymbol{x_t}$ at any arbitrary timestep $\boldsymbol{t}$ can be derived based on $\boldsymbol{x_0}$ and $\boldsymbol{\beta_t}$:

$$q(x_t \mid x_0) = N\left(x_t; \sqrt{\bar{\alpha}_t}x_0, \sqrt{1-\bar{\alpha}_t}I\right)(3)$$

$$\bar{\alpha}_t = \prod_{i=1}^{t}(1-\beta_i)\,(4)$$

Content and Technical Route

DDPM: Reverse Process

Purpose: Recreate a sample in the specific data distribution $\boldsymbol{q(x_0)}$ from sampling Gaussian noise $\boldsymbol{x_t}$.

$$q(\, x_t \mid x_{t-1} \,) \rightarrow q(\, x_{t-1} \mid x_t \,) \qquad \textit{Hard to Estimate!}$$

Solution: use a U-Net θ to approximate these conditional probabilities by fitting the mean and variance.

The reverse Gaussian transition:

$$p_\theta(\, x_{t-1} \mid x_t \,) = N\big(x_{t-1};\ \mu_\theta\big(x_t, t\big), \Sigma_\theta(x_t, t)\big)(5)$$

Substituting (1) and (3) into the conditional probability $q(\, x_{t-1} \mid x_t, x_0 \,) \sim N\big(x_{t-1};\ \tilde{\mu}_t\big(x_t, x_0\big), \tilde{\beta}_t I\big)$, the mean $\tilde{\mu}_t\big(x_t, x_0\big)$ and variance $\tilde{\beta}_t$ can be parameterized with Bayes' rule:

$$\tilde{\mu}_t\big(x_t, x_0\big) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\right)(6) \qquad \alpha_t = 1 - \beta_t$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t(7)$$

Content and Technical Route

DDPM: Optimization

Optimization Objective: Recreate a sampling distribution $q(\tilde{x}_0)$ close to the prior data distribution $\boldsymbol{q(x_0)}$**,** which can be achieved by minimizing the negative log-likelihood (NLL) and optimized by using the **variational lower bound**:

$$\begin{aligned} -\log p_\theta(\mathrm{x}_0) \leq & -\log p_\theta(\mathrm{x}_0) \\ & + D_{KL}(q(x_{1:T} \mid x_0) \| p_\theta(x_{1:T} \mid x_0)) \\ & = \mathbb{E}_q\left[log \frac{q(\mathrm{x}_{1:T} \mid \mathrm{x}_0)}{p_\theta(x_{0:T})}\right] \\ & = \mathbb{E}_q\left[-\log p(\mathrm{x}_T) - \sum_{t \geq 1} \log \frac{p_\theta(\mathrm{x}_{t-1} \mid \mathrm{x}_t)}{q(\mathrm{x}_t \mid \mathrm{x}_{t-1})}\right] \\ & = \mathbb{E}_q[\underbrace{D_{KL}(q(x_T \mid x_0) \| p(x_T))}_{\mathcal{L}_T} \\ & + \sum_{t>1} \underbrace{D_{KL}(q(x_{t-1} \mid x_t, x_0) \| p_\theta(x_{t-1} \mid x_t))}_{\mathcal{L}_{t-1}} \\ & \underbrace{-\log p_\theta(x_0 \mid x_1)]}_{\mathcal{L}_0} \end{aligned} \quad (8)$$

where L_T and L_0 are fixed values after the data distribution x_0 and the noise scheme β are determined. The parameterized L_{t-1} can be calculated by substituting the mean and variance of the $q(x_{t-1} \mid x_t, x_0)$ and $p_\theta(x_{t-1} \mid x_t)$:

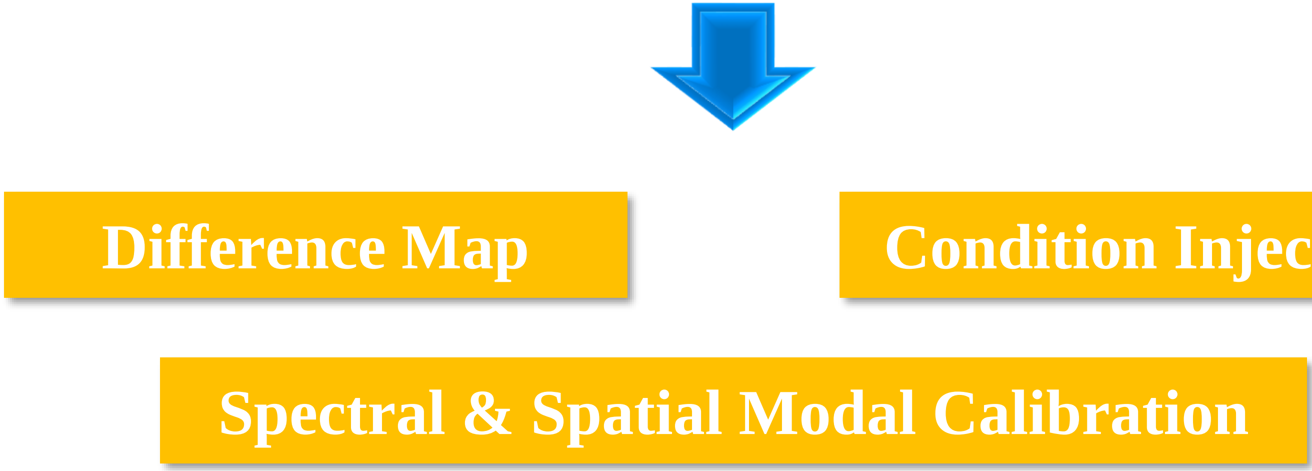
$$L_{t-1} = E_{x_0,\epsilon}\left[\frac{1}{2\sigma^2}\left\|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\right\|^2\right] (9)$$

Content and Technical Route

Model Design

Questions:

- ◆ **How to destroy HRMS with rich spatial and spectral information into approximate Gaussian noise x_T in limited timesteps?**
- ◆ **How to guide the random Gaussian noise x_T simulate the process of HRMS reconstruction with high uncertainty?**



Difference Map

Condition Injection

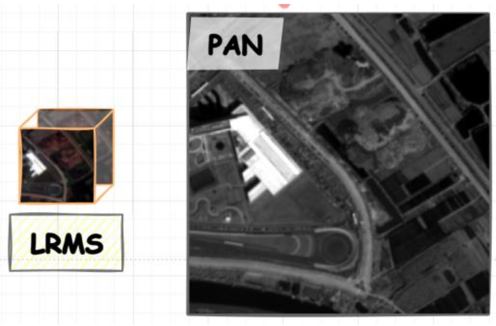
Spectral & Spatial Modal Calibration

Content and Technical Route

Difference Map

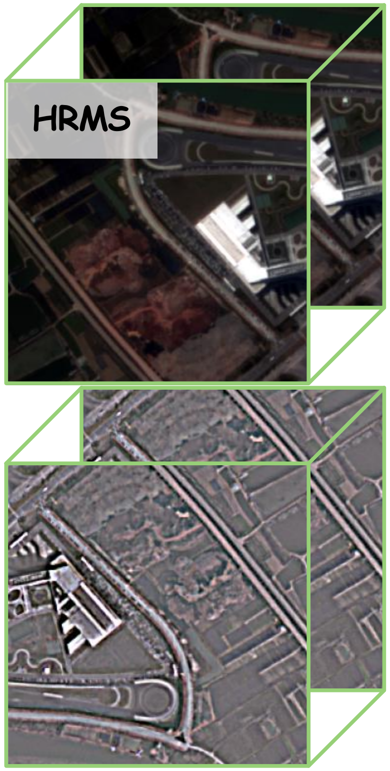
Two Considerations:

- **Effectively alleviate the difficulty of the work that converting the HRMS into a Gaussian noise and reconstructing it by reversion in a limited number of timesteps.**
- **The fusion objective of PanDiff is more clearly defined, which undoubtedly leads to better performance of the model.**



$$HRMS = F_{\theta}(LRMS, PAN)$$

$$\Delta MS = F_{\theta}(LRMS, PAN)$$



Content and Technical Route

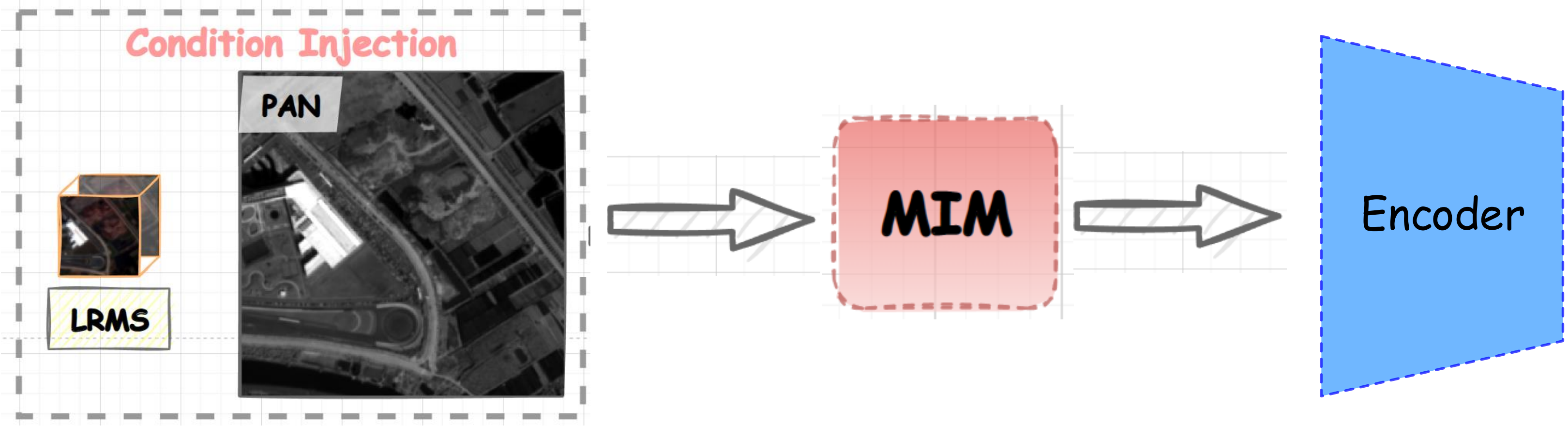
Condition Injection

Using PAN and LRMS images as condition injections to control the reverse process of reconstructing:

$$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1} \mid x_t, cond)\,(10)$$

$$cond = \Phi(PAN, LRMS)(11)$$

where $\Phi(\cdot)$ is the encoder branch for processing PAN and LRMS images as the injected condition



Content and Technical Route

MIM

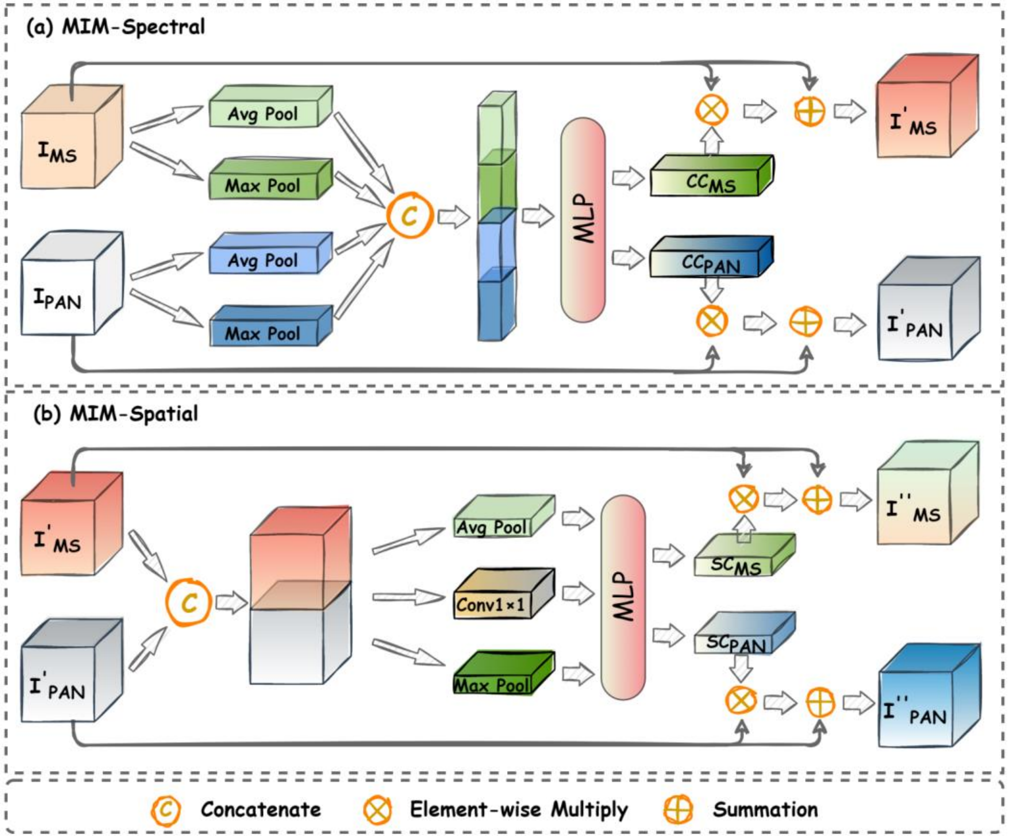
Significant modal differences exist between PAN and LRMS images, allowing PAN and LRMS images to guide the modeling of $q(x_{t-1} \mid x_t)$ in neural networks by focusing on various aspects.

➢ MIM-Spectral

1. Averaging & Max global pooling via channel dimension
2. Multi layer perceptron (MLP)
3. Multiply the weights

➢ MIM-Spatial

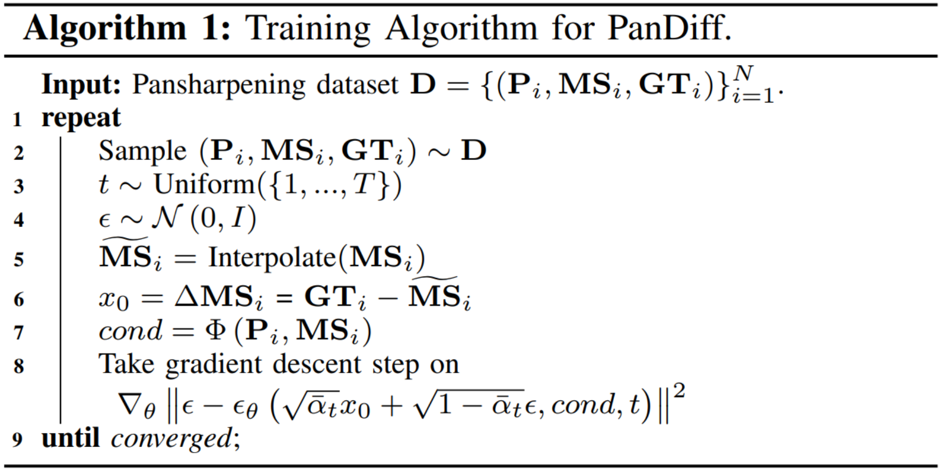
1. Averaging & Max global pooling in channel dimension
2. Conv1×1
3. MLP
4. Multiply the weights



Content and Technical Route

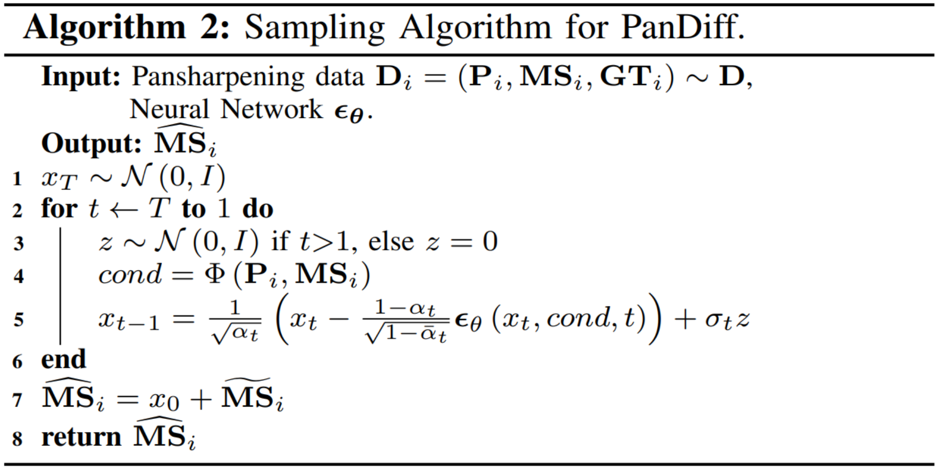
Algorithms

PanDiff Training



```
Algorithm 1: Training Algorithm for PanDiff.
Input: Pansharpening dataset $\mathbf{D} = \{(\mathbf{P}_i, \mathbf{MS}_i, \mathbf{GT}_i)\}_{i=1}^{N}$.
repeat
    Sample $(\mathbf{P}_i, \mathbf{MS}_i, \mathbf{GT}_i) \sim \mathbf{D}$
    $t \sim \text{Uniform}(\{1, ..., T\})$
    $\epsilon \sim \mathcal{N}(0, I)$
    $\widetilde{\mathbf{MS}}_i = \text{Interpolate}(\mathbf{MS}_i)$
    $x_0 = \Delta\mathbf{MS}_i = \mathbf{GT}_i - \widetilde{\mathbf{MS}}_i$
    $cond = \Phi(\mathbf{P}_i, \mathbf{MS}_i)$
    Take gradient descent step on
      $\nabla_\theta \left\| \epsilon - \epsilon_\theta \left( \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, cond, t \right) \right\|^2$
until converged;
```

PanDiff Sampling

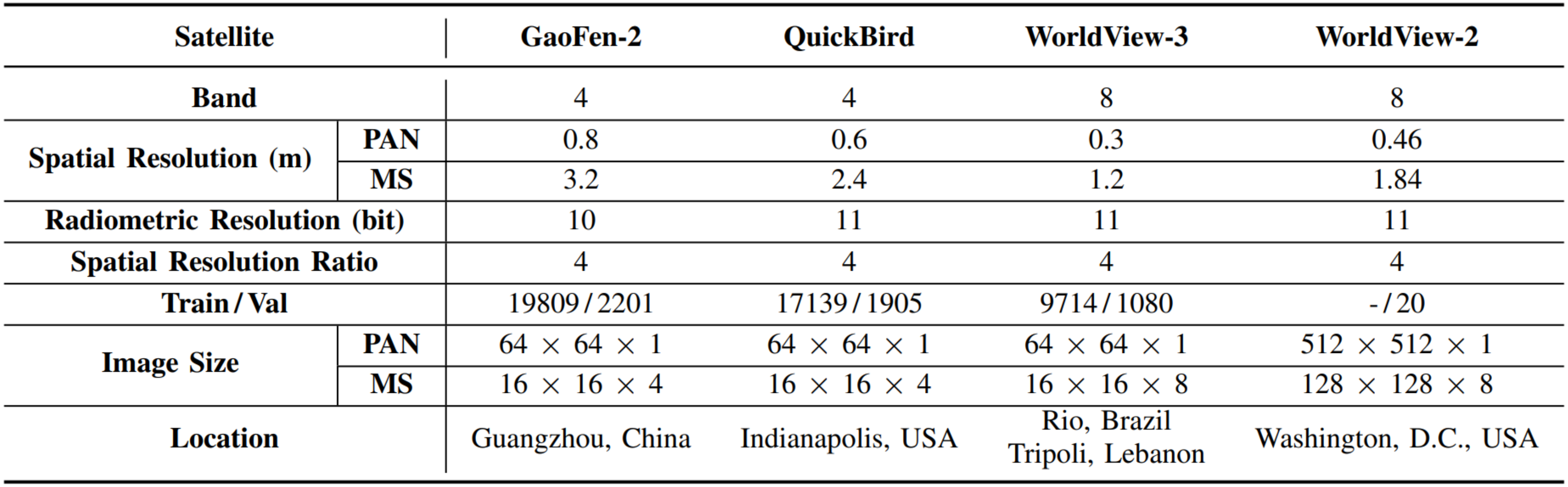


```
Algorithm 2: Sampling Algorithm for PanDiff.
Input: Pansharpening data $\mathbf{D}_i = (\mathbf{P}_i, \mathbf{MS}_i, \mathbf{GT}_i) \sim \mathbf{D}$,
       Neural Network $\boldsymbol{\epsilon_\theta}$.
Output: $\widehat{\mathbf{MS}}_i$
$x_T \sim \mathcal{N}(0, I)$
for $t \leftarrow T$ to 1 do
    $z \sim \mathcal{N}(0, I)$ if $t>1$, else $z = 0$
    $cond = \Phi(\mathbf{P}_i, \mathbf{MS}_i)$
    $x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left( x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(x_t, cond, t) \right) + \sigma_t z$
end
$\widehat{\mathbf{MS}}_i = x_0 + \widetilde{\mathbf{MS}}_i$
return $\widehat{\mathbf{MS}}_i$
```

Experiment Results

Datasets

The PanCollection dataset containing data from four satellites (GaoFen-2, QuickBird, WorldView-3, and WorldView-2) is utilized to evaluate PanDiff with other state-of-the-art methods fairly and comprehensively.



Satellite		GaoFen-2	QuickBird	WorldView-3	WorldView-2
Band		4	4	8	8
Spatial Resolution (m)	**PAN**	0.8	0.6	0.3	0.46
	MS	3.2	2.4	1.2	1.84
Radiometric Resolution (bit)		10	11	11	11
Spatial Resolution Ratio		4	4	4	4
Train / Val		19809 / 2201	17139 / 1905	9714 / 1080	- / 20
Image Size	**PAN**	64 × 64 × 1	64 × 64 × 1	64 × 64 × 1	512 × 512 × 1
	MS	16 × 16 × 4	16 × 16 × 4	16 × 16 × 8	128 × 128 × 8
Location		Guangzhou, China	Indianapolis, USA	Rio, Brazil Tripoli, Lebanon	Washington, D.C., USA

Experiment Results

Experiment Details

Benchmarks

CS-based Methods: BT-H, and **BDSD-PC**

MRA-based Methods: MTF-GLP-FS, and **MTF-GLP-HPM-R**

DL-based Methods:

- **CNN: PNN, PanNet, DRPNN, MSDCNN, DiCNN, SSconv**, and **TDNet**
- **GAN: PSGAN**, and **MDSSC-GAN**

Evaluation Metrics

Reduced Resolution: PSNR, SSIM, SAM, ERGAS, and **SCC**

Full Resolution: D_S, D_{lambda}, **QNR** and **HQNR**

Experiment Results

Experiment Settings

Experimental Parameter Setting

- AdamW is chosen as the optimizer, the total number of training times $iter_0$ is 320,000, the initial learning rate lr_0 is 0.0001, and the learning rate lr is updated using the MultiStep learning rate scheduler, the batch size is 384.

Data Preprocessing and Augmentation

Data Preprocessing

- **Data Normalization:** both the input and output of DDPM need to be approximated as standard Gaussian distributions.

$$d' = 2 \times \frac{d}{2\gamma} - 1(12)$$

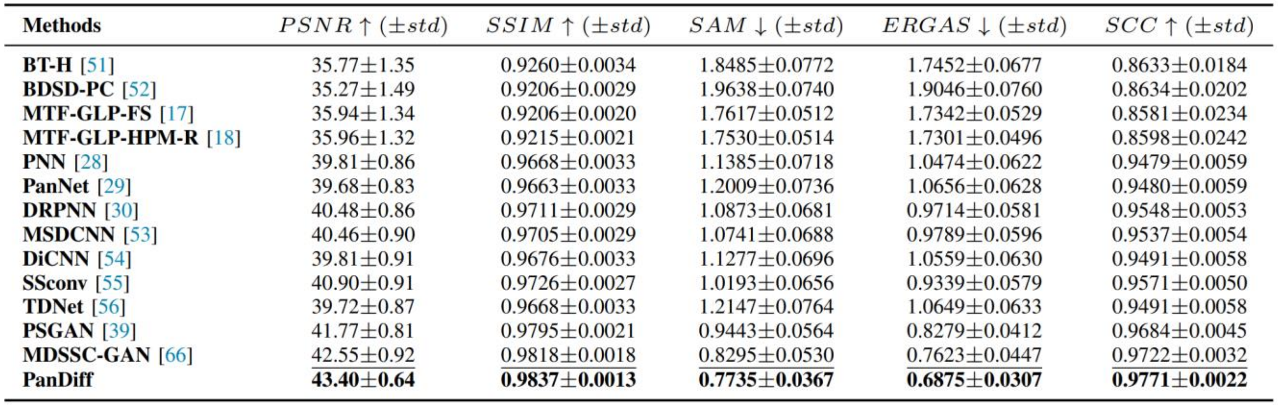
Data Augmentation

- **Random Flipping:** Includes vertical flip and horizontal flip;
- **Random Rotation:** Includes 90°, 180, and 270 rotations.

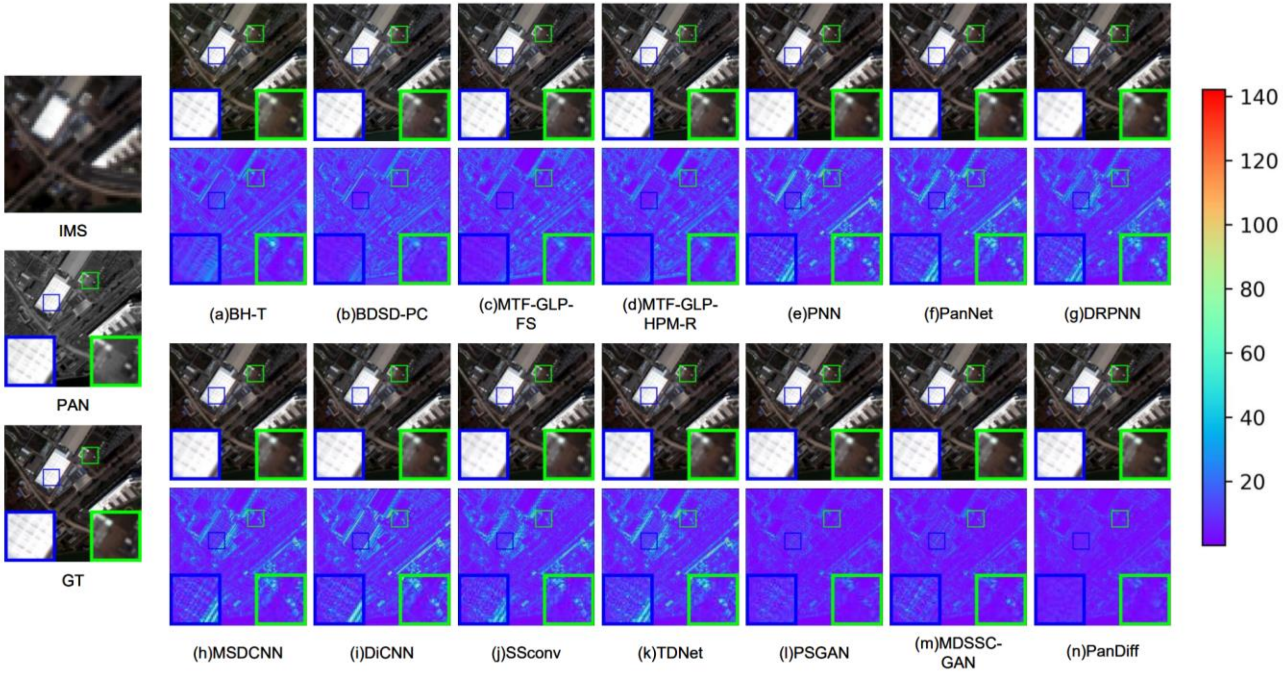
Experiment Results

Reduced Resolution

GF-2



Methods	$PSNR\uparrow(\pm std)$	$SSIM\uparrow(\pm std)$	$SAM\downarrow(\pm std)$	$ERGAS\downarrow(\pm std)$	$SCC\uparrow(\pm std)$
BT-H [51]	35.77±1.35	0.9260±0.0034	1.8485±0.0772	1.7452±0.0677	0.8633±0.0184
BDSD-PC [52]	35.27±1.49	0.9206±0.0029	1.9638±0.0740	1.9046±0.0760	0.8634±0.0202
MTF-GLP-FS [17]	35.94±1.34	0.9206±0.0020	1.7617±0.0512	1.7342±0.0529	0.8581±0.0234
MTF-GLP-HPM-R [18]	35.96±1.32	0.9215±0.0021	1.7530±0.0514	1.7301±0.0496	0.8598±0.0242
PNN [28]	39.81±0.86	0.9668±0.0033	1.1385±0.0718	1.0474±0.0622	0.9479±0.0059
PanNet [29]	39.68±0.83	0.9663±0.0033	1.2009±0.0736	1.0656±0.0628	0.9480±0.0059
DRPNN [30]	40.48±0.86	0.9711±0.0029	1.0873±0.0681	0.9714±0.0581	0.9548±0.0053
MSDCNN [53]	40.46±0.90	0.9705±0.0029	1.0741±0.0688	0.9789±0.0596	0.9537±0.0054
DiCNN [54]	39.81±0.91	0.9676±0.0033	1.1277±0.0696	1.0559±0.0630	0.9491±0.0058
SSconv [55]	40.90±0.91	0.9726±0.0027	1.0193±0.0656	0.9339±0.0579	0.9571±0.0050
TDNet [56]	39.72±0.87	0.9668±0.0033	1.2147±0.0764	1.0649±0.0633	0.9491±0.0058
PSGAN [39]	41.77±0.81	0.9795±0.0021	0.9443±0.0564	0.8279±0.0412	0.9684±0.0045
MDSSC-GAN [66]	42.55±0.92	0.9818±0.0018	0.8295±0.0530	0.7623±0.0447	0.9722±0.0032
PanDiff	**43.40±0.64**	**0.9837±0.0013**	**0.7735±0.0367**	**0.6875±0.0307**	**0.9771±0.0022**



IMS

PAN

GT

(a)BH-T (b)BDSD-PC (c)MTF-GLP-FS (d)MTF-GLP-HPM-R (e)PNN (f)PanNet (g)DRPNN

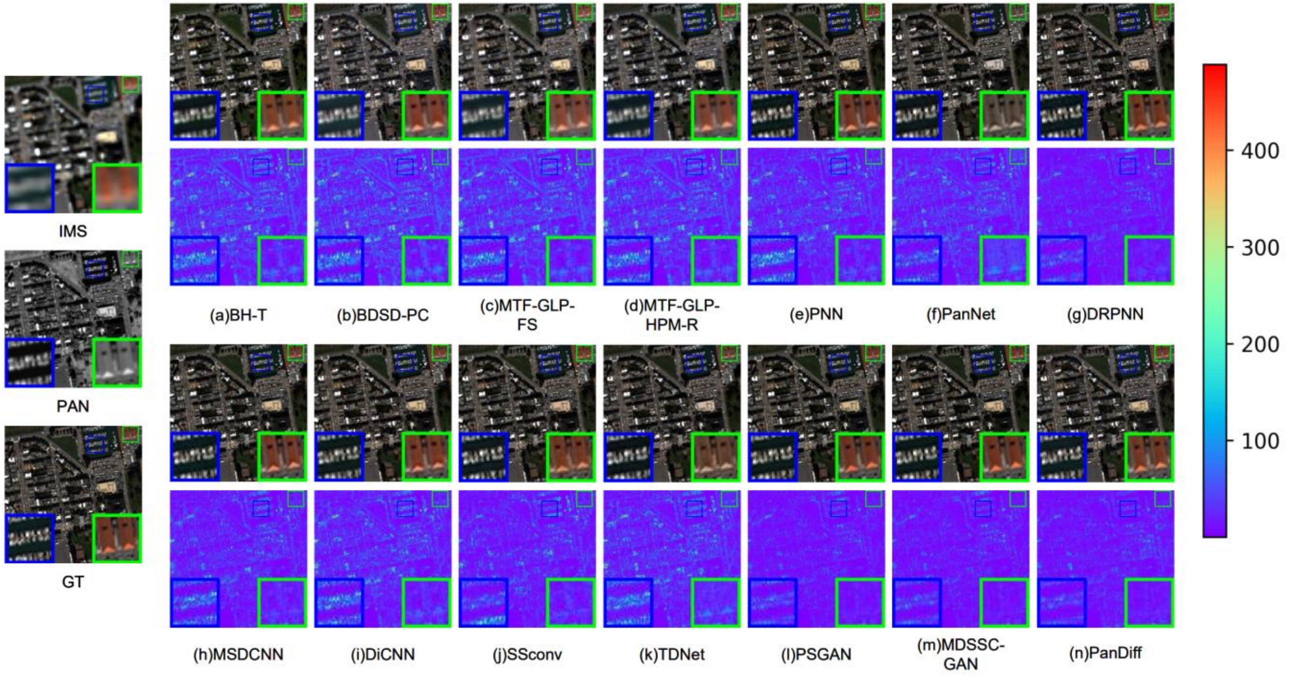
(h)MSDCNN (i)DiCNN (j)SSconv (k)TDNet (l)PSGAN (m)MDSSC-GAN (n)PanDiff

Experiment Results

Reduced Resolution

QuickBird

Methods	$PSNR \uparrow (\pm std)$	$SSIM \uparrow (\pm std)$	$SAM \downarrow (\pm std)$	$ERGAS \downarrow (\pm std)$	$SCC \uparrow (\pm std)$
BT-H [51]	35.61±0.90	0.8940±0.0058	6.3700±0.3679	7.0396±1.3166	0.8109±0.1796
BDSD-PC [52]	35.95±0.60	0.8957±0.0045	6.7232±0.3074	6.3606±0.0745	0.8979±0.0032
MTF-GLP-FS [17]	36.12±0.64	0.8967±0.0055	6.4589±0.3403	6.2240±0.0841	0.8975±0.0035
MTF-GLP-HPM-R [18]	36.12±0.65	0.8985±0.0059	6.4546±0.3811	6.7141±1.0089	0.8650±0.0070
PNN [28]	37.69±0.82	0.9289±0.0057	5.1577±0.2658	5.3945±0.3255	0.9411±0.0105
PanNet [29]	37.92±0.85	0.9321±0.0069	5.0604±0.2593	5.2545±0.3729	0.9511±0.0090
DRPNN [30]	39.31±0.73	0.9494±0.0051	4.5977±0.2165	4.4963±0.3218	0.9661±0.0066
MSDCNN [53]	38.62±0.78	0.9410±0.0054	4.8659±0.2411	4.8606±0.3183	0.9560±0.0083
DiCNN [54]	37.55±0.88	0.9266±0.0060	5.1528±0.2750	5.4922±0.3181	0.9370±0.0104
SSconv [55]	38.85±0.78	0.9433±0.0059	4.7277±0.2329	4.7828±0.3686	0.9627±0.0076
TDNet [56]	37.50±0.85	0.9266±0.0065	5.2085±0.2707	5.5289±0.3581	0.9451±0.0096
PSGAN [39]	40.07±0.75	0.9565±0.0047	4.2570±0.2025	4.1415±0.3126	0.9718±0.0058
MDSSC-GAN [66]	40.01±0.76	0.9557±0.0049	**4.2450±0.1998**	4.1772±0.3145	0.9710±0.0058
PanDiff	**41.70±0.73**	**0.9569±0.0073**	4.3193±0.2553	**3.7824±0.3526**	**0.9725±0.0067**

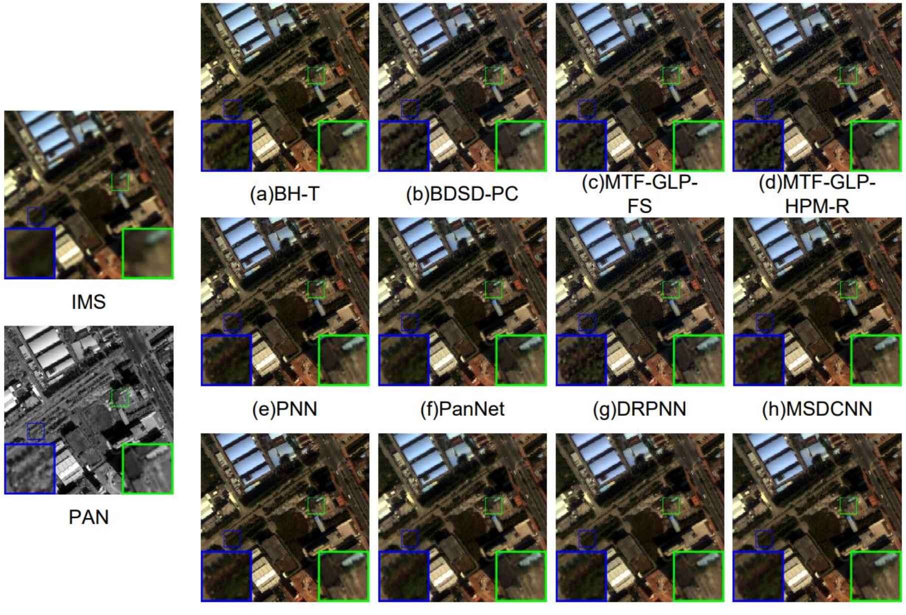


Experiment Results

Full Resolution

GF-2

Methods	$D_\lambda \downarrow (\pm std)$	$D_S \downarrow (\pm std)$	$QNR \uparrow (\pm std)$	$HQNR \uparrow (\pm std)$
BT-H [51]	0.0891±0.0335	0.1712±0.0388	0.7399±0.0551	0.7558±0.0567
BDSD-PC [52]	0.0926±0.0292	0.1652±0.0362	0.7767±0.0539	0.7582±0.0509
MTF-GLP-FS [17]	0.0370±0.0138	0.1539±0.0351	0.7636±0.0542	0.8150±0.0404
MTF-GLP-HPM-R [18]	0.0364±0.0131	0.1531±0.0353	0.7650±0.0545	0.8163±0.0400
PNN [28]	0.0490±0.0693	0.1263±0.0338	0.8256±0.0194	0.8236±0.0564
PanNet [29]	0.0353±0.0105	0.1035±0.0258	0.8494±0.0409	0.8649±0.0271
DRPNN [30]	0.0374±0.0148	0.1115±0.0321	0.8265±0.0519	0.8555±0.0390
MSDCNN [53]	0.0298±0.0118	0.0869±0.0194	0.8729±0.0323	0.8859±0.0202
DiCNN [54]	0.0329±0.0098	0.0921±0.0248	0.8580±0.0394	0.8781±0.0273
SSconv [55]	<u>0.0228±0.0084</u>	<u>0.0478±0.0156</u>	<u>0.9232±0.0274</u>	<u>0.9304±0.0146</u>
TDNet [56]	0.0301±0.0096	0.0839±0.0202	0.8786±0.0324	0.8885±0.0201
PanDiff	**0.0223±0.0103**	**0.0323±0.0131**	**0.9396±0.0250**	**0.9461±0.0125**



IMS

PAN

(a)BH-T

(b)BDSD-PC

(c)MTF-GLP-FS

(d)MTF-GLP-HPM-R

(e)PNN

(f)PanNet

(g)DRPNN

(h)MSDCNN

(i)DiCNN

(j)SSconv

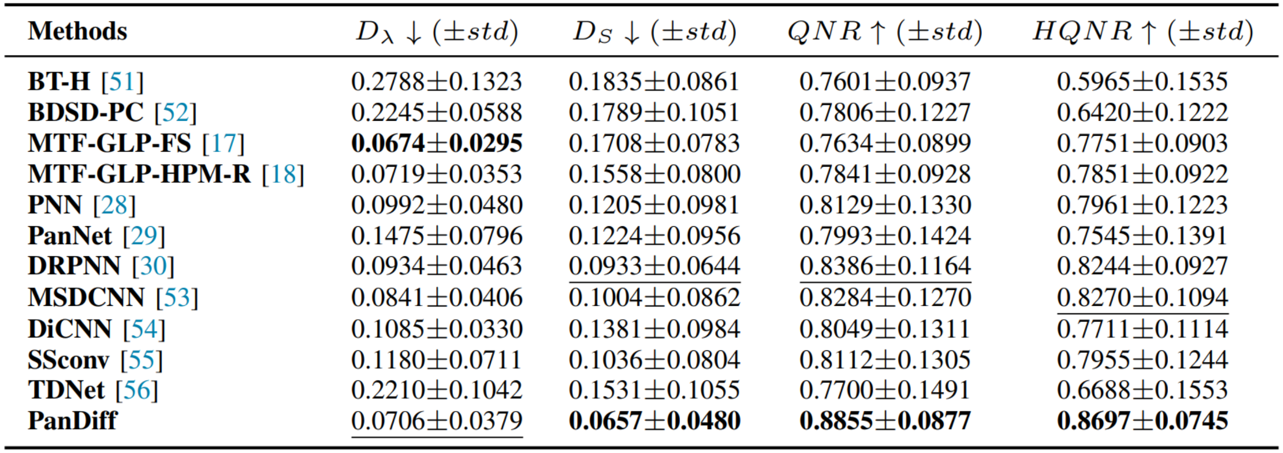
(k)TDNet

(m)PanDiff

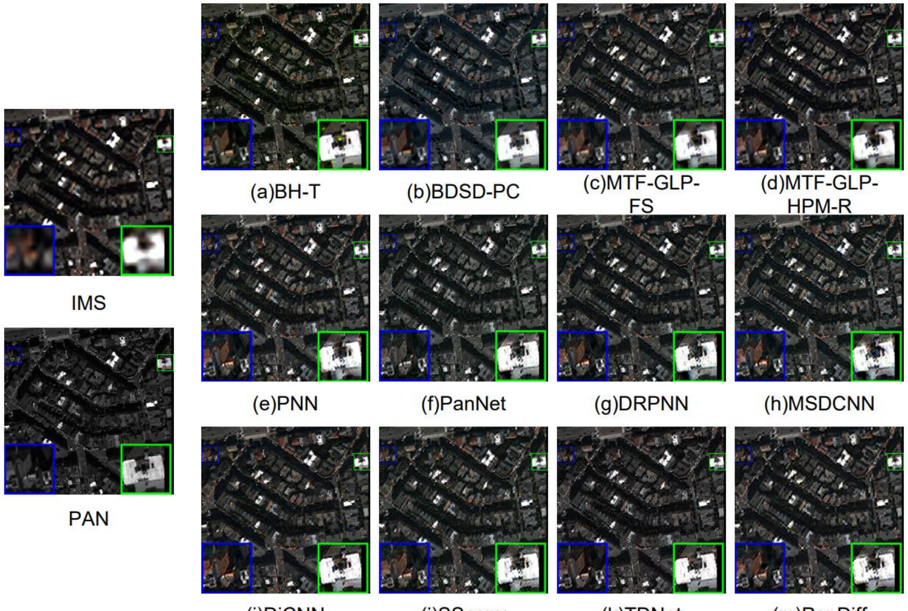
Experiment Results

Full Resolution

QuickBird



Methods	$D_\lambda \downarrow (\pm std)$	$D_S \downarrow (\pm std)$	$QNR \uparrow (\pm std)$	$HQNR \uparrow (\pm std)$
BT-H [51]	0.2788±0.1323	0.1835±0.0861	0.7601±0.0937	0.5965±0.1535
BDSD-PC [52]	0.2245±0.0588	0.1789±0.1051	0.7806±0.1227	0.6420±0.1222
MTF-GLP-FS [17]	**0.0674±0.0295**	0.1708±0.0783	0.7634±0.0899	0.7751±0.0903
MTF-GLP-HPM-R [18]	0.0719±0.0353	0.1558±0.0800	0.7841±0.0928	0.7851±0.0922
PNN [28]	0.0992±0.0480	0.1205±0.0981	0.8129±0.1330	0.7961±0.1223
PanNet [29]	0.1475±0.0796	0.1224±0.0956	0.7993±0.1424	0.7545±0.1391
DRPNN [30]	0.0934±0.0463	0.0933±0.0644	0.8386±0.1164	0.8244±0.0927
MSDCNN [53]	0.0841±0.0406	0.1004±0.0862	0.8284±0.1270	0.8270±0.1094
DiCNN [54]	0.1085±0.0330	0.1381±0.0984	0.8049±0.1311	0.7711±0.1114
SSconv [55]	0.1180±0.0711	0.1036±0.0804	0.8112±0.1305	0.7955±0.1244
TDNet [56]	0.2210±0.1042	0.1531±0.1055	0.7700±0.1491	0.6688±0.1553
PanDiff	0.0706±0.0379	**0.0657±0.0480**	**0.8855±0.0877**	**0.8697±0.0745**



IMS

PAN

(a)BH-T

(b)BDSD-PC

(c)MTF-GLP-FS

(d)MTF-GLP-HPM-R

(e)PNN

(f)PanNet

(g)DRPNN

(h)MSDCNN

(i)DiCNN

(j)SSconv

(k)TDNet

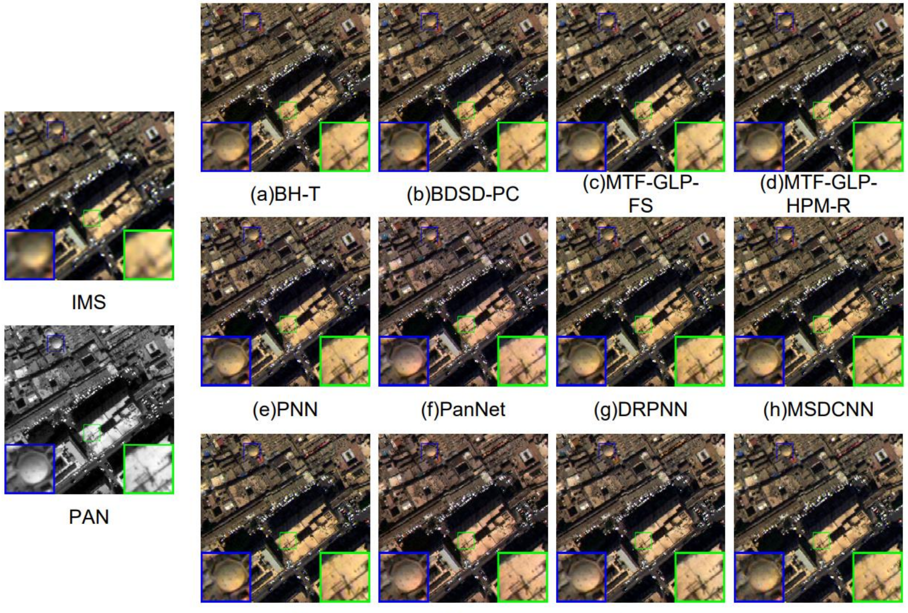
(m)PanDiff

Experiment Results

Full Resolution

WorldView-3

Methods	$D_\lambda \downarrow (\pm std)$	$D_S \downarrow (\pm std)$	$QNR \uparrow (\pm std)$	$HQNR \uparrow (\pm std)$
BT-H [51]	0.1851±0.1848	0.1493±0.0937	0.7568±0.1678	0.7080±0.2157
BDSD-PC [52]	0.1505±0.1204	0.1464±0.1159	0.7901±0.1713	0.7375±0.1860
MTF-GLP-FS [17]	**0.0631±0.0579**	0.1390±0.1191	0.7833±0.1781	0.8126±0.1524
MTF-GLP-HPM-R [18]	0.0635±0.0575	0.1370±0.1179	0.7870±0.1752	0.8139±0.1507
PNN [28]	0.1160±0.1086	0.0667±0.0210	0.8374±0.1184	0.8262±0.1121
PanNet [29]	0.1862±0.1886	0.0721±0.0263	0.8292±0.1023	0.7579±0.1849
DRPNN [30]	0.1157±0.1163	0.0903±0.0712	0.8086±0.1498	0.8107±0.1528
MSDCNN [53]	0.1105±0.1119	0.0761±0.0466	0.8341±0.1297	0.8258±0.1350
DiCNN [54]	0.1023±0.0977	0.0724±0.0435	0.8373±0.1307	0.8356±0.1196
SSconv [55]	0.2021±0.2147	0.0925±0.0592	0.8021±0.1386	0.7350±0.2268
TDNet [56]	0.2116±0.2183	0.1043±0.0950	0.7961±0.1627	0.7222±0.2401
PanDiff	0.0982±0.1096	**0.0537±0.0467**	**0.9091±0.0742**	**0.8571±0.1336**



IMS

PAN

(a)BH-T (b)BDSD-PC (c)MTF-GLP-FS (d)MTF-GLP-HPM-R

(e)PNN (f)PanNet (g)DRPNN (h)MSDCNN

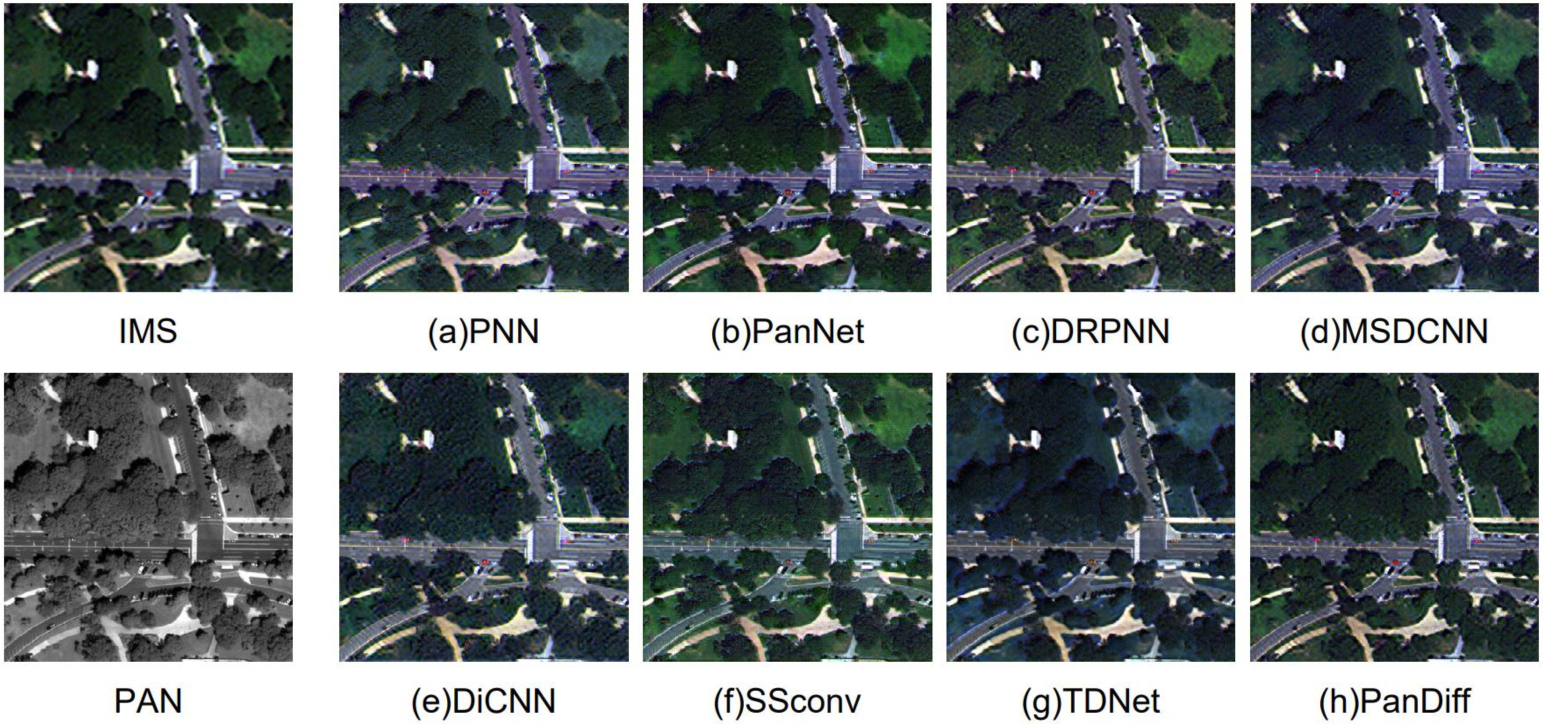
(i)DiCNN (j)SSconv (k)TDNet (m)PanDiff

Experiment Results

Generalization Test

Using WorldView-2 images to perform cross-sensor, cross-resolution generalization experiments on the model trained with WorldView-3 data.

PanDiff shows high robustness with excellent spectral retention and spatial enhancement capabilities.



IMS

(a)PNN

(b)PanNet

(c)DRPNN

(d)MSDCNN

PAN

(e)DiCNN

(f)SSconv

(g)TDNet

(h)PanDiff

Experiment Results

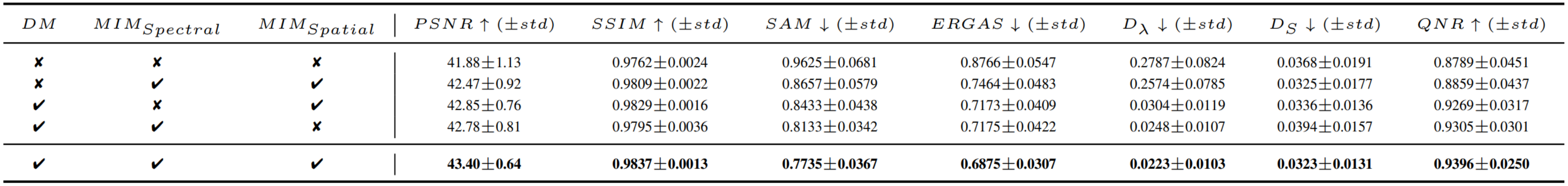
Ablation Study

Effectiveness of Difference Map

In the reduced resolution experiments, the results are as expected; because more data information must be reconstructed, not using the DM decreases performance by 0.93, 0.0028, 0.0922, and 0.0589 for PSNR, SSIM, SAM, and ERGAS, respectively. However, omitting the DM sharply degrades the model's capacity to retain spectral information for full-resolution images, although the difference in spatial detail retention ability is insignificant.

Effectiveness of MIM

PanDiff without MIM-Spectral has a considerable reduction in the spectral metrics SAM and D_λ, 0.0698 and 0.0081, respectively; PanDiff with missing MIM-Spatial has a reduction in the spatial structure metrics SSIM and D_S , 0.0042 and 0.0071, respectively.



DM	$MIM_{Spectral}$	$MIM_{Spatial}$	$PSNR \uparrow (\pm std)$	$SSIM \uparrow (\pm std)$	$SAM \downarrow (\pm std)$	$ERGAS \downarrow (\pm std)$	$D_\lambda \downarrow (\pm std)$	$D_S \downarrow (\pm std)$	$QNR \uparrow (\pm std)$
✘	✘	✘	41.88±1.13	0.9762±0.0024	0.9625±0.0681	0.8766±0.0547	0.2787±0.0824	0.0368±0.0191	0.8789±0.0451
✘	✔	✔	42.47±0.92	0.9809±0.0022	0.8657±0.0579	0.7464±0.0483	0.2574±0.0785	0.0325±0.0177	0.8859±0.0437
✔	✘	✔	42.85±0.76	0.9829±0.0016	0.8433±0.0438	0.7173±0.0409	0.0304±0.0119	0.0336±0.0136	0.9269±0.0317
✔	✔	✘	42.78±0.81	0.9795±0.0036	0.8133±0.0342	0.7175±0.0422	0.0248±0.0107	0.0394±0.0157	0.9305±0.0301
✔	✔	✔	**43.40±0.64**	**0.9837±0.0013**	**0.7735±0.0367**	**0.6875±0.0307**	**0.0223±0.0103**	**0.0323±0.0131**	**0.9396±0.0250**

Experiment Results

Conclusion

- PanDiff is a generative model based on the DDPM which is first designed for pansharpening;
- PanDiff *changes the learning objective* of the traditional fusion networks. It decomposes the complex fusion process of PAN and LRMS images into a multi-step Markov process, and actually learns the data distribution of the difference map (DM) of HRMS and interpolated MS (IMS), rather than the spatial and spectral information of HRMS；
- PanDiff no longer treats the input PAN and MS as the object of feature extraction, it *injects* the PAN and MS images intercalibrated by a modal intercalibration module (MIM) as *conditions* to guide the U-Net to learn the data distribution of the DM of HRMS and IMS；
- Comparisons between PanDiff with other state-of-the-art methods on GaoFen-2, QuickBird, and WorldView-3 data show the *significant effectiveness* and *superiority* of PanDiff.

Thanks !