Linear Attention
meets
Semantic Segmentation

汇报人：李睿

汇报人

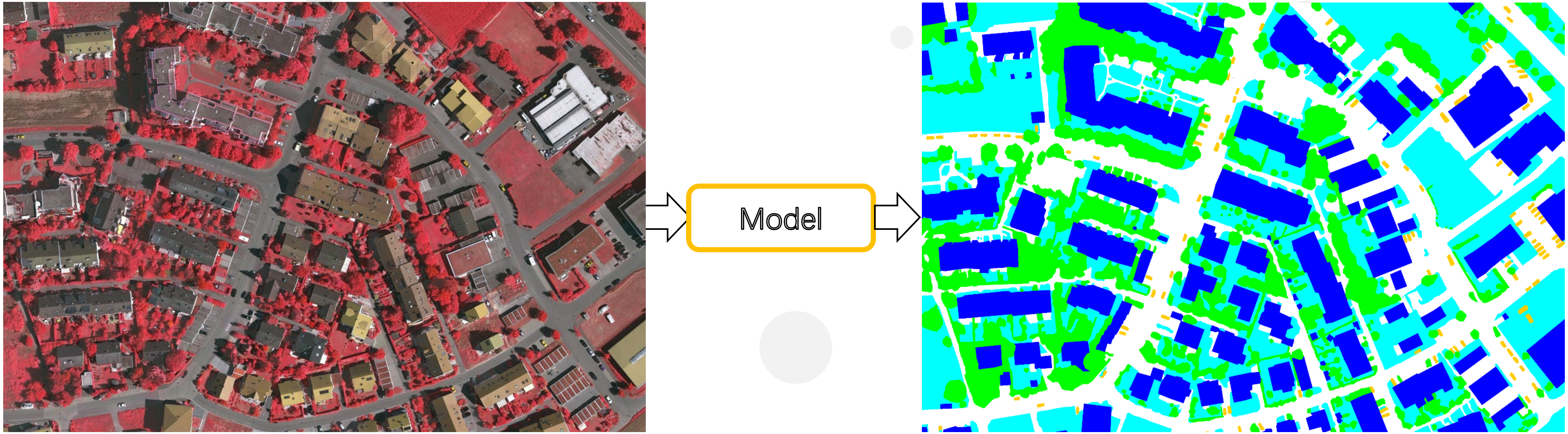
李睿

计算机视觉
语义分割
深度学习

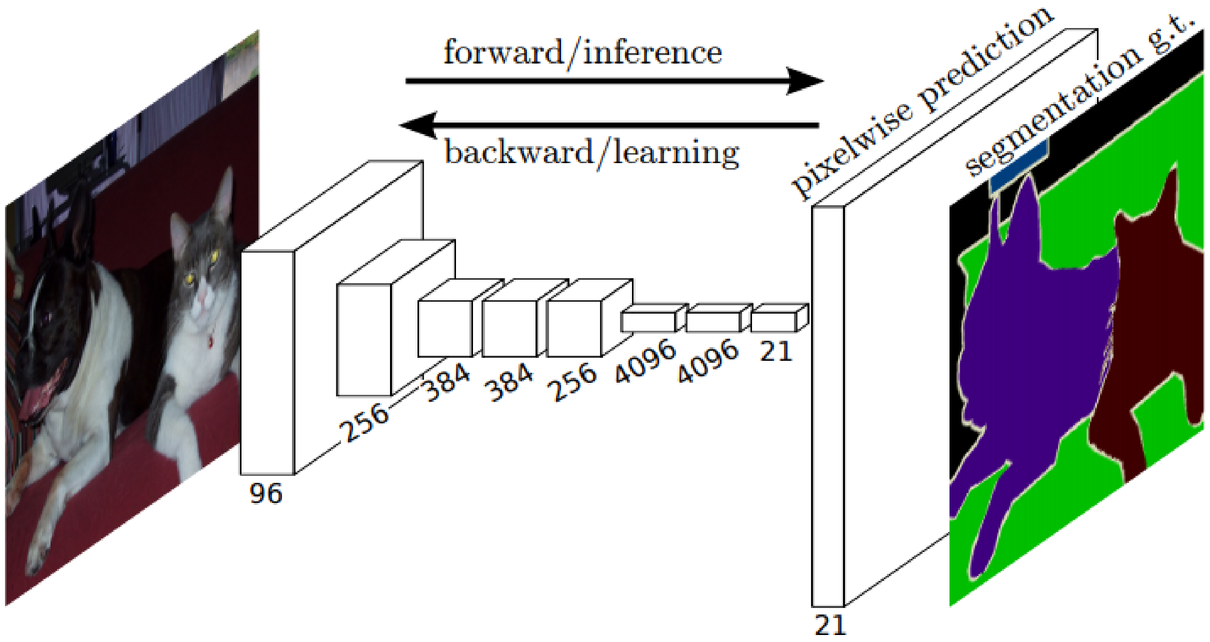
01

Semantic Segmentation

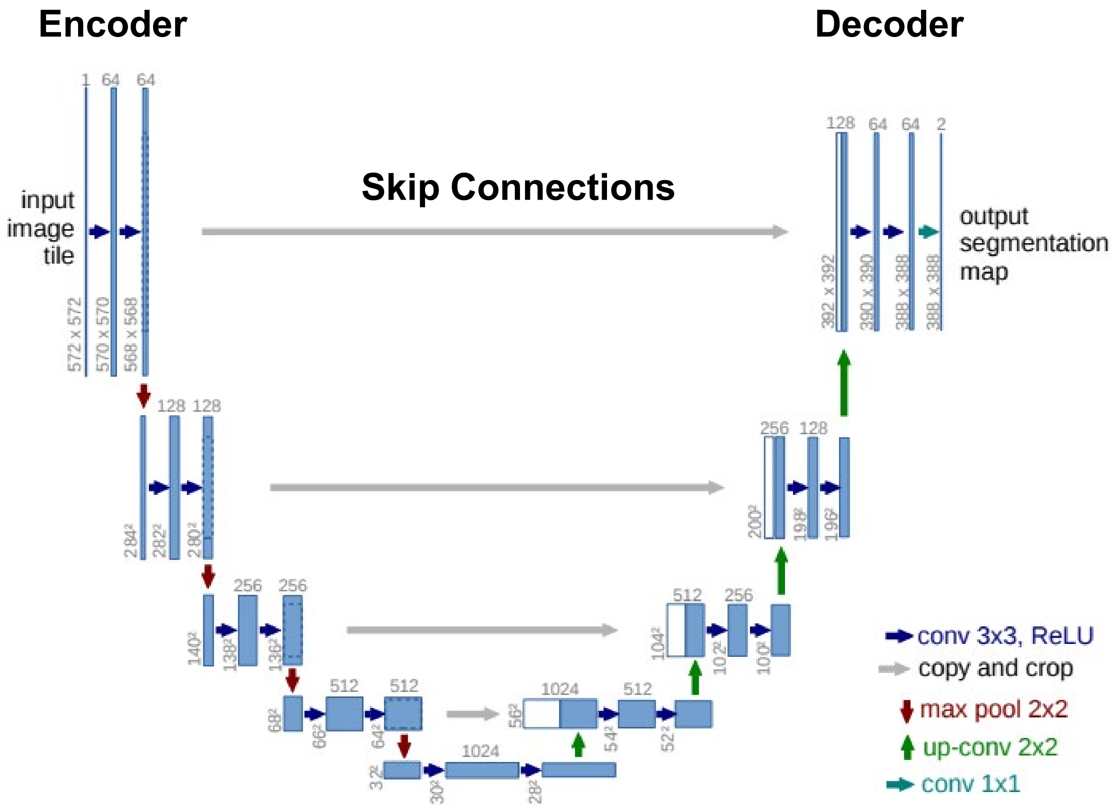
1 Semantic Segmentation



1 Semantic Segmentation

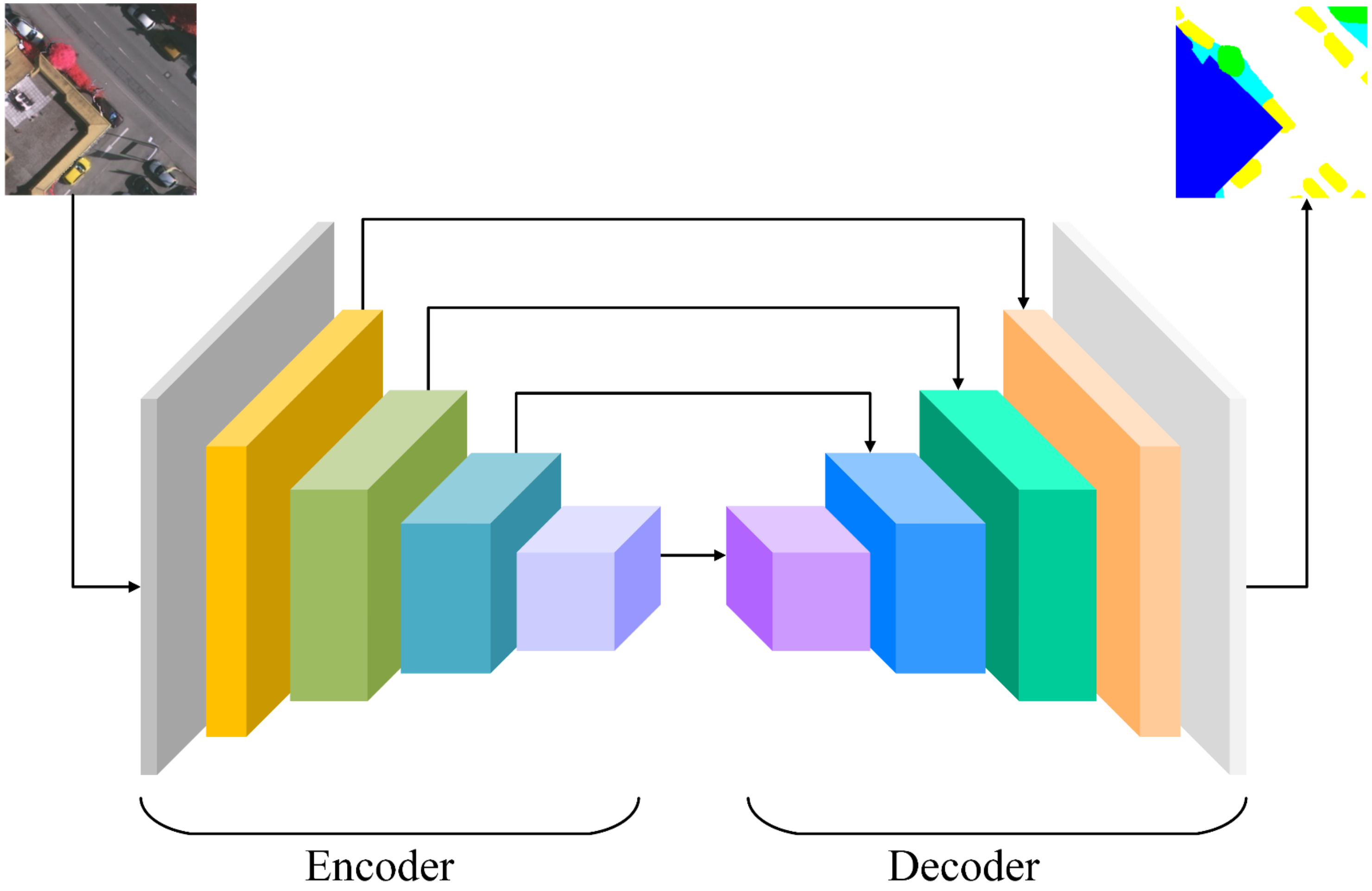


FCN



U-Net

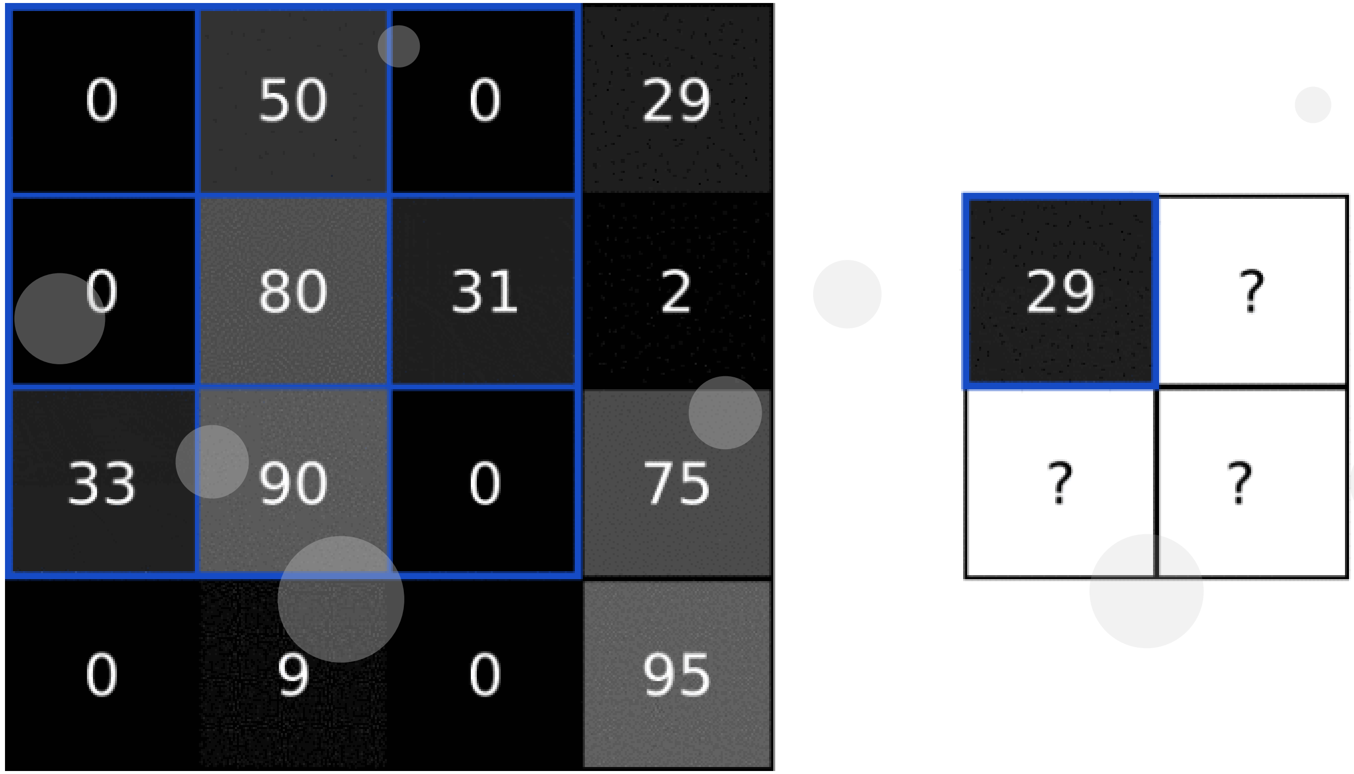
1 Semantic Segmentation



02

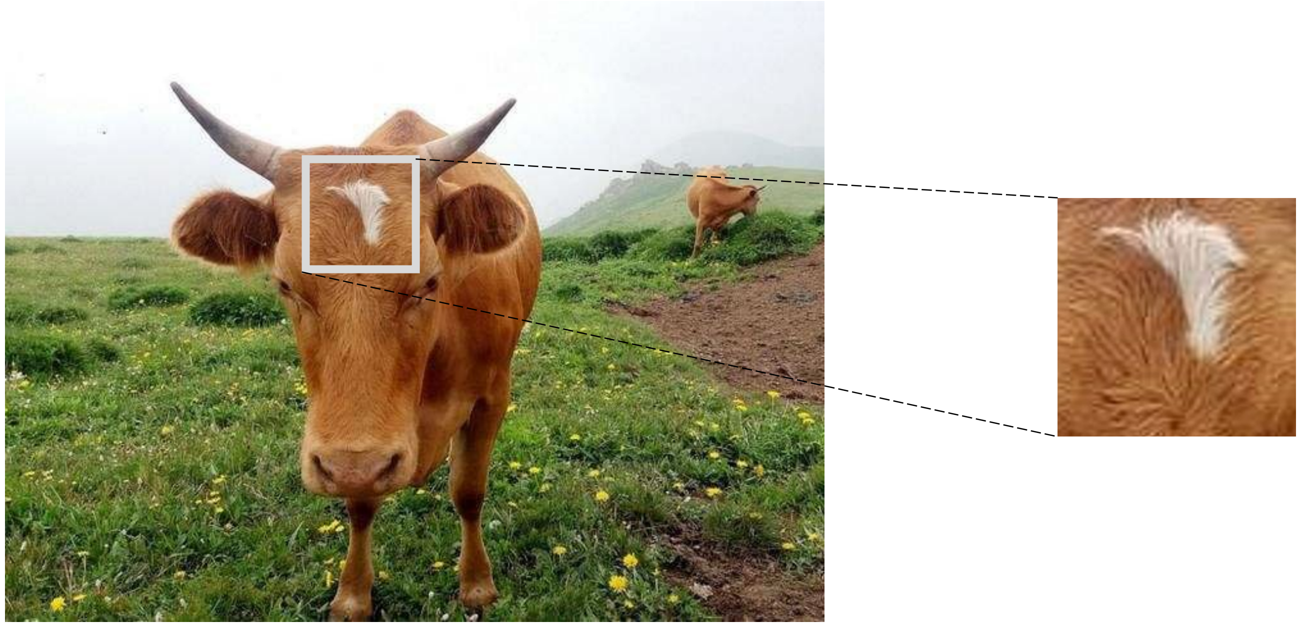
Contextual Information

2 Contextual Information



卷积操作->局部图像特征
池化操作->局部特征筛选

2 Contextual Information



Contextual Information

局部特征存在模糊性和不确定性

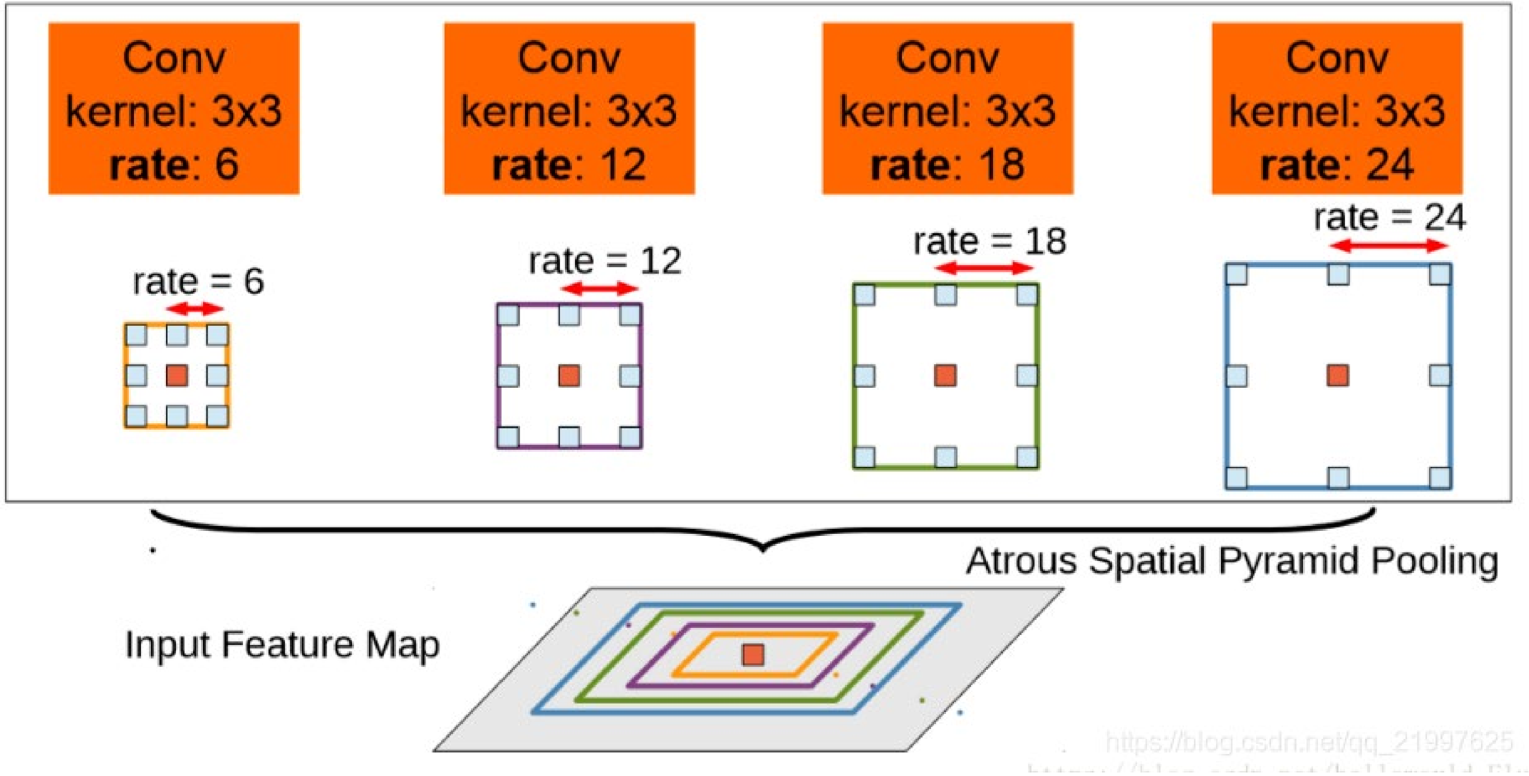
03

Solutions

3 Solutions

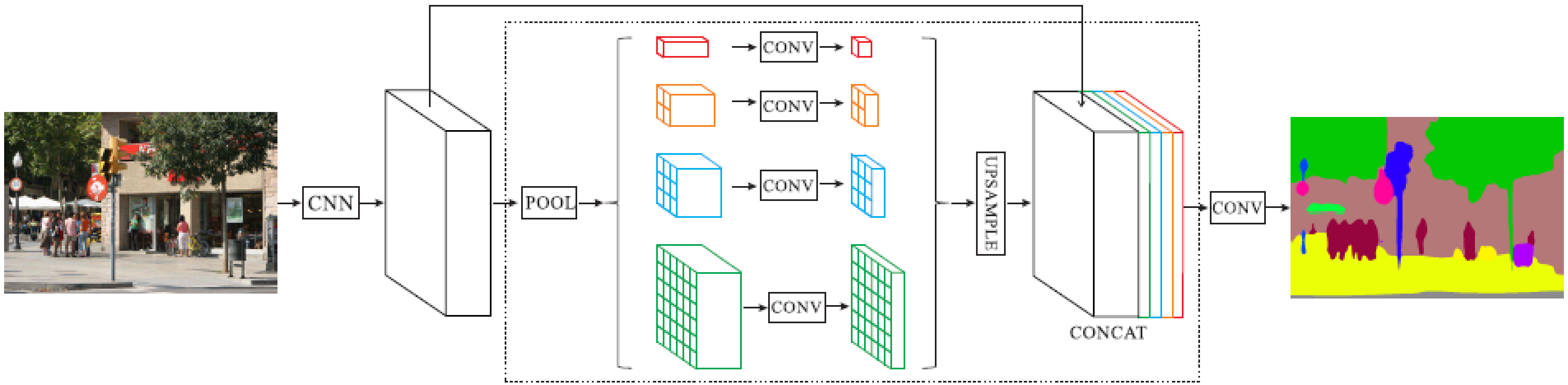
ASPP

- atrous spatial pyramid pooling
- L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” arXiv preprint arXiv:1706.05587, 2017



PPM

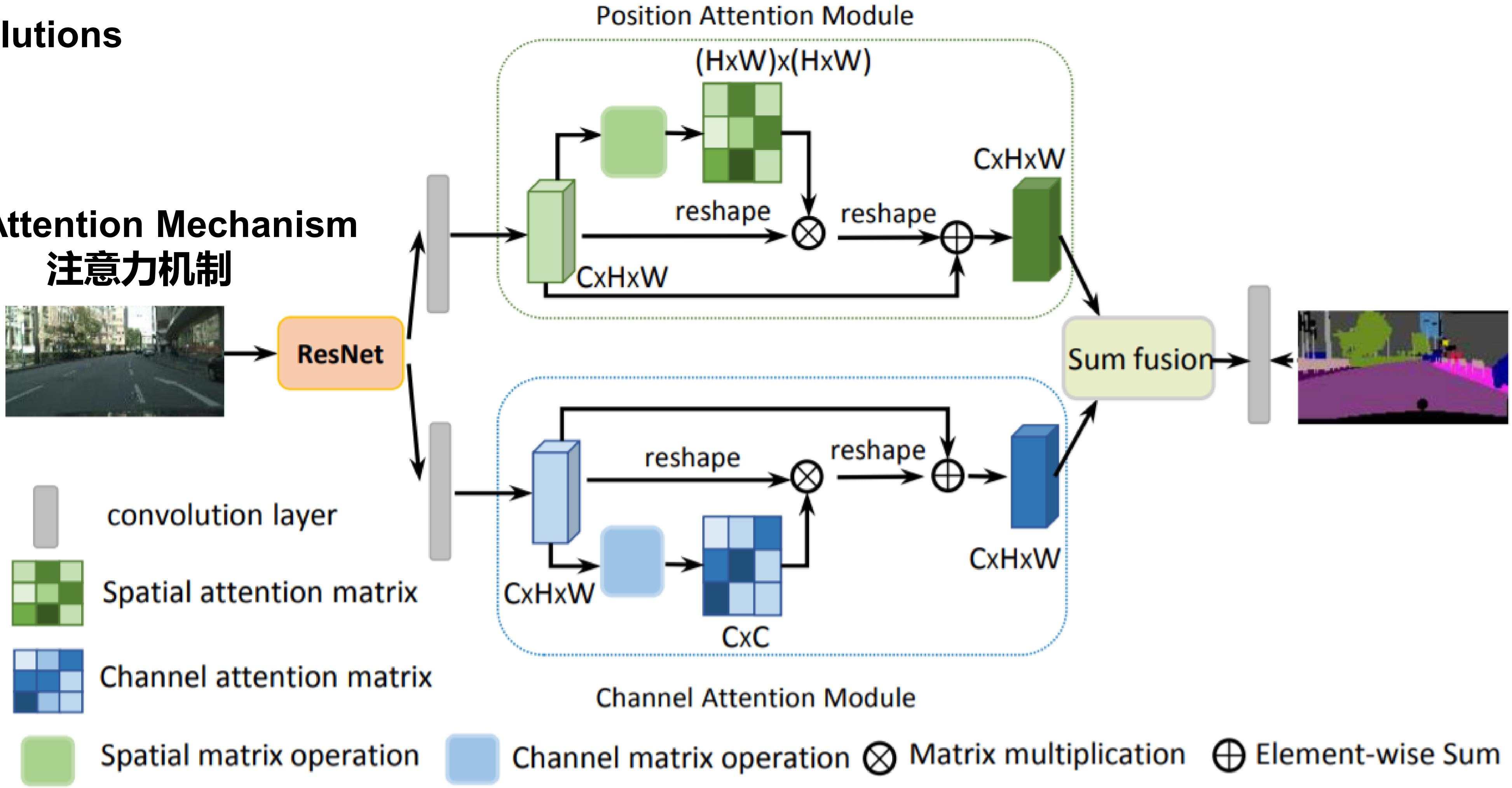
- **pyramid pooling module**
- H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2881–2890.



(a) Input Image (b) Feature Map (c) Pyramid Pooling Module (d) Final Prediction

3 Solutions

Self-Attention Mechanism
注意力机制



- Wang X, Girshick R, Gupta A, et al. Non-local neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7794-7803.
- Fu J, Liu J, Tian H, et al. Dual attention network for scene segmentation[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 3146-3154.

04

Limitation

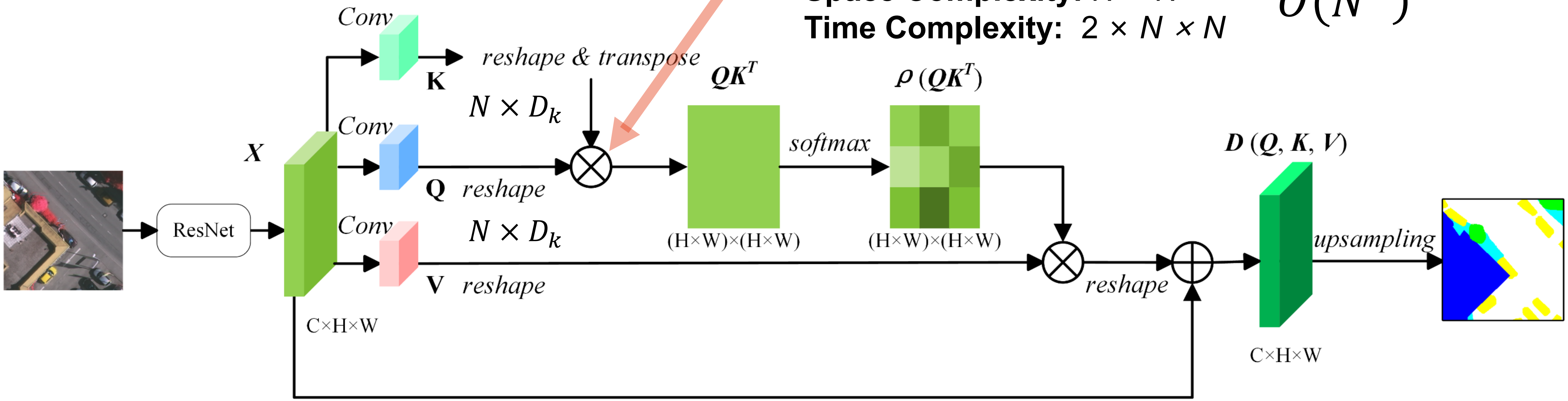
4 Limitation

$$\underbrace{N \times D_k}_{Q} \times \underbrace{D_k \times N}_{K^T} \rightarrow N \times N$$

Space Complexity: $N \times N$

Time Complexity: $2 \times N \times N$

$$O(N^2)$$



Projected Matrices

$$\begin{cases} \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_k \in \mathbb{R}^{N \times D_k}; \\ \boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_q \in \mathbb{R}^{N \times D_k}; \\ \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_v \in \mathbb{R}^{N \times D_v}. \end{cases}$$

$N = H \times W$

H: Height

W: Width

Calculating the Similarities

$$\rho(\boldsymbol{Q}\boldsymbol{K}^T) = softmax_{row}(\boldsymbol{Q}\boldsymbol{K}^T)$$

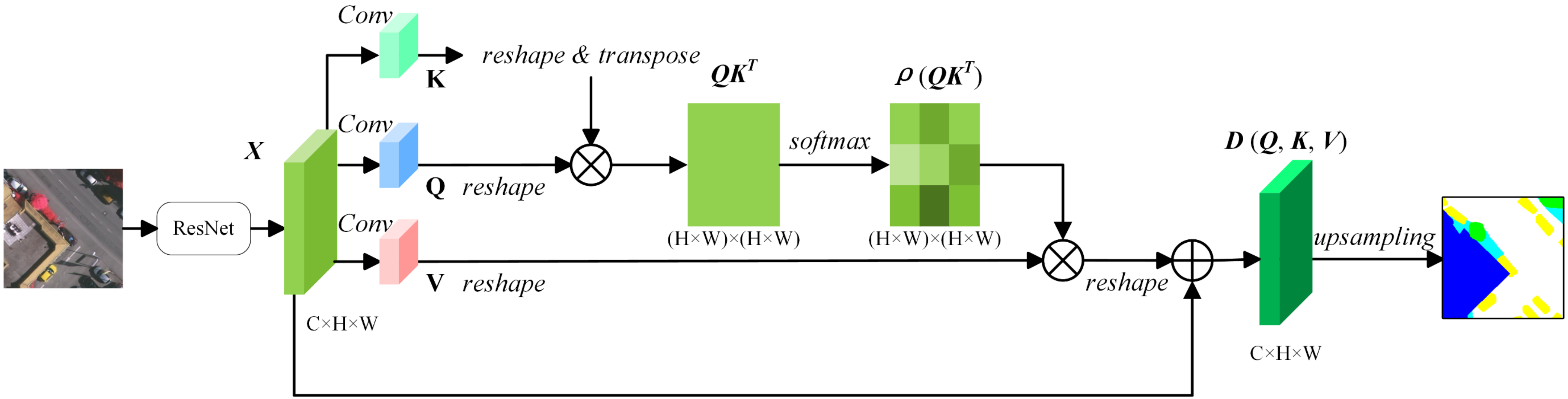
Output

$$D(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \rho(\boldsymbol{Q}\boldsymbol{K}^T)\boldsymbol{V}$$

05

Simplification

5 Simplification



两种方案

避免softmax操作

$$D(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \rho(\boldsymbol{Q}\boldsymbol{K}^T)\boldsymbol{V}$$

$$\rho(\boldsymbol{Q}\boldsymbol{K}^T) = \boxed{softmax_{row}}(\boldsymbol{Q}\boldsymbol{K}^T)$$

5 Simplification

方案1

一般化推广

$$\begin{cases} \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_k \in \mathbb{R}^{N\times D_k}; \\ \boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_q \in \mathbb{R}^{N\times D_k}; \\ \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_v \in \mathbb{R}^{N\times D_v}. \end{cases}$$

$$D(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \rho(\boldsymbol{Q}\boldsymbol{K}^T)\boldsymbol{V}$$

$$\rho(\boldsymbol{Q}\boldsymbol{K}^T) = \boxed{softmax_{row}}(\boldsymbol{Q}\boldsymbol{K}^T)$$

EQ1: $$D(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})_i = \frac{\sum_{j=1}^{N} e^{\boldsymbol{q}_i{}^T\boldsymbol{k}_j}\boldsymbol{v}_j}{\sum_{j=1}^{N} e^{\boldsymbol{q}_i{}^T\boldsymbol{k}_j}}$$

EQ2:

$$D(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})_i = \frac{\sum_{j=1}^{N} sim(\boldsymbol{q}_i,\boldsymbol{k}_j)\boldsymbol{v}_j}{\sum_{j=1}^{N} sim(\boldsymbol{q}_i,\boldsymbol{k}_j)}, sim(\boldsymbol{q}_i,\boldsymbol{k}_j) \geq 0.$$

$$\mathrm{sim}(\boldsymbol{q}_i,\boldsymbol{k}_j) = \phi(\boldsymbol{q}_i)^T\varphi(\boldsymbol{k}_j)$$

EQ3: $$D(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})_i = \frac{\sum_{j=1}^{N}\phi(\boldsymbol{q}_i)^T\varphi(\boldsymbol{k}_j)\boldsymbol{v}_j}{\sum_{j=1}^{N}\phi(\boldsymbol{q}_i)^T\varphi(\boldsymbol{k}_j)}$$

EQ4: $$D(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})_i = \frac{\phi(\boldsymbol{q}_i)^T\sum_{j=1}^{N}\varphi(\boldsymbol{k}_j)\boldsymbol{v}_j{}^T}{\phi(\boldsymbol{q}_i)^T\sum_{j=1}^{N}\varphi(\boldsymbol{k}_j)}$$

EQ5: $$D(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \frac{\phi(\boldsymbol{Q})\varphi(\boldsymbol{K})^T\boldsymbol{V}}{\phi(\boldsymbol{Q})\sum_j\varphi(\boldsymbol{K})^T_{i,j}}$$

5 Simplification

方案2

泰勒展开

$$D(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \rho(\boldsymbol{Q}\boldsymbol{K}^T)\boldsymbol{V}$$

$$\rho(\boldsymbol{Q}\boldsymbol{K}^T) = \boxed{softmax_{row}}(\boldsymbol{Q}\boldsymbol{K}^T)$$

EQ1: $$D(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})_i = \frac{\sum_{j=1}^{N} e^{\boldsymbol{q}_i{}^T\boldsymbol{k}_j}\boldsymbol{v}_j}{\sum_{j=1}^{N} e^{\boldsymbol{q}_i{}^T\boldsymbol{k}_j}}$$

$$e^{\boldsymbol{q}_i{}^T\boldsymbol{k}_j} \approx 1 + \boldsymbol{q}_i{}^T\boldsymbol{k}_j$$

EQ2:

$$D(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})_i = \frac{\sum_{j=1}^{N}\left(1+\left(\frac{\boldsymbol{q}_i}{\|\boldsymbol{q}_i\|_2}\right)^T\left(\frac{\boldsymbol{k}_j}{\|\boldsymbol{k}_j\|_2}\right)\right)\boldsymbol{v}_j}{\sum_{j=1}^{N}\left(1+\left(\frac{\boldsymbol{q}_i}{\|\boldsymbol{q}_i\|_2}\right)^T\left(\frac{\boldsymbol{k}_j}{\|\boldsymbol{k}_j\|_2}\right)\right)}$$

EQ3:

$$D(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})_i = \frac{\sum_{j=1}^{N}\boldsymbol{v}_j + \left(\frac{\boldsymbol{q}_i}{\|\boldsymbol{q}_i\|_2}\right)^T\sum_{j=1}^{N}\left(\frac{\boldsymbol{k}_j}{\|\boldsymbol{k}_j\|_2}\right)\boldsymbol{v}_j^T}{N + \left(\frac{\boldsymbol{q}_i}{\|\boldsymbol{q}_i\|_2}\right)^T\sum_{j=1}^{N}\left(\frac{\boldsymbol{k}_j}{\|\boldsymbol{k}_j\|_2}\right)}$$

EQ4: $$D(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \frac{\sum_j \boldsymbol{V}_{i,j} + \left(\frac{\boldsymbol{Q}}{\|\boldsymbol{Q}\|_2}\right)\left(\left(\frac{\boldsymbol{K}}{\|\boldsymbol{K}\|_2}\right)^T\boldsymbol{V}\right)}{N + \left(\frac{\boldsymbol{Q}}{\|\boldsymbol{Q}\|_2}\right)\sum_j\left(\frac{\boldsymbol{K}}{\|\boldsymbol{K}\|_2}\right)^T_{i,j}}$$

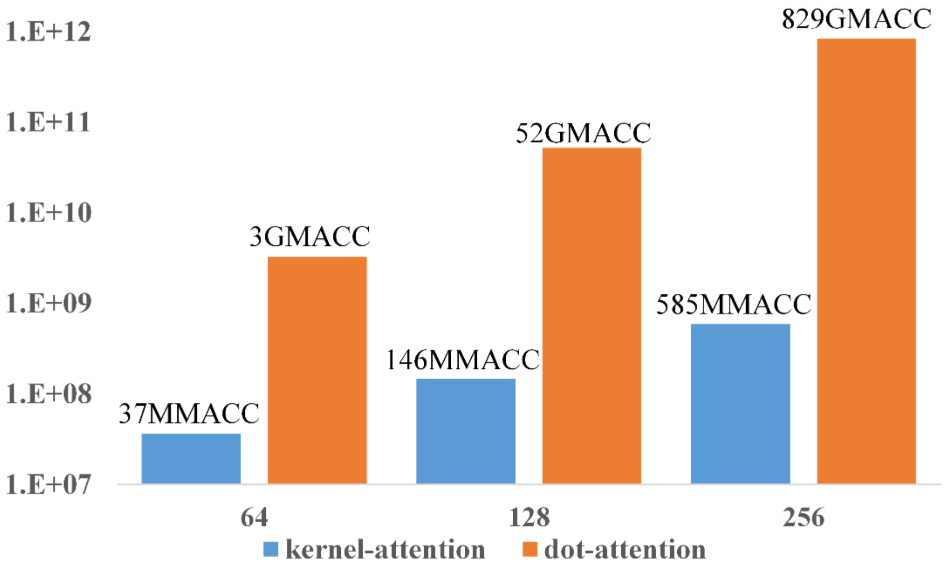
06

Results

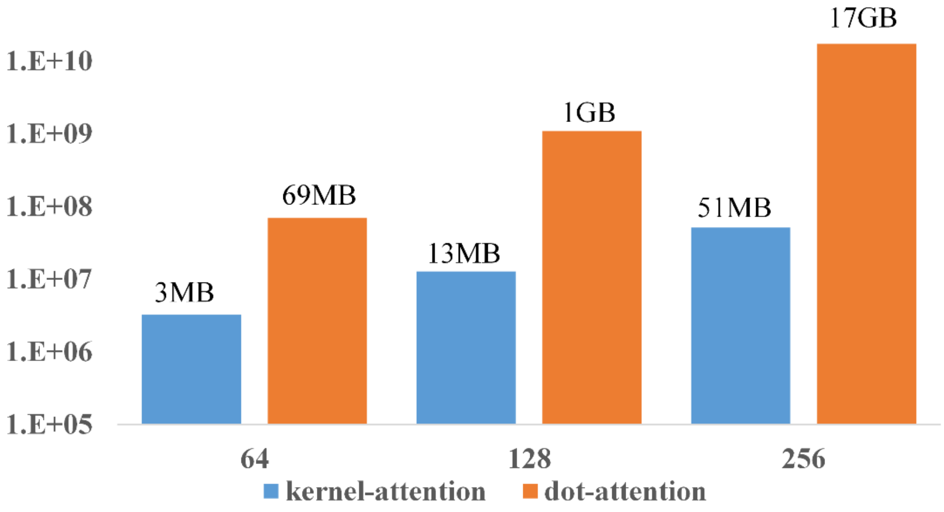
6 Results

方案1

一般化推广



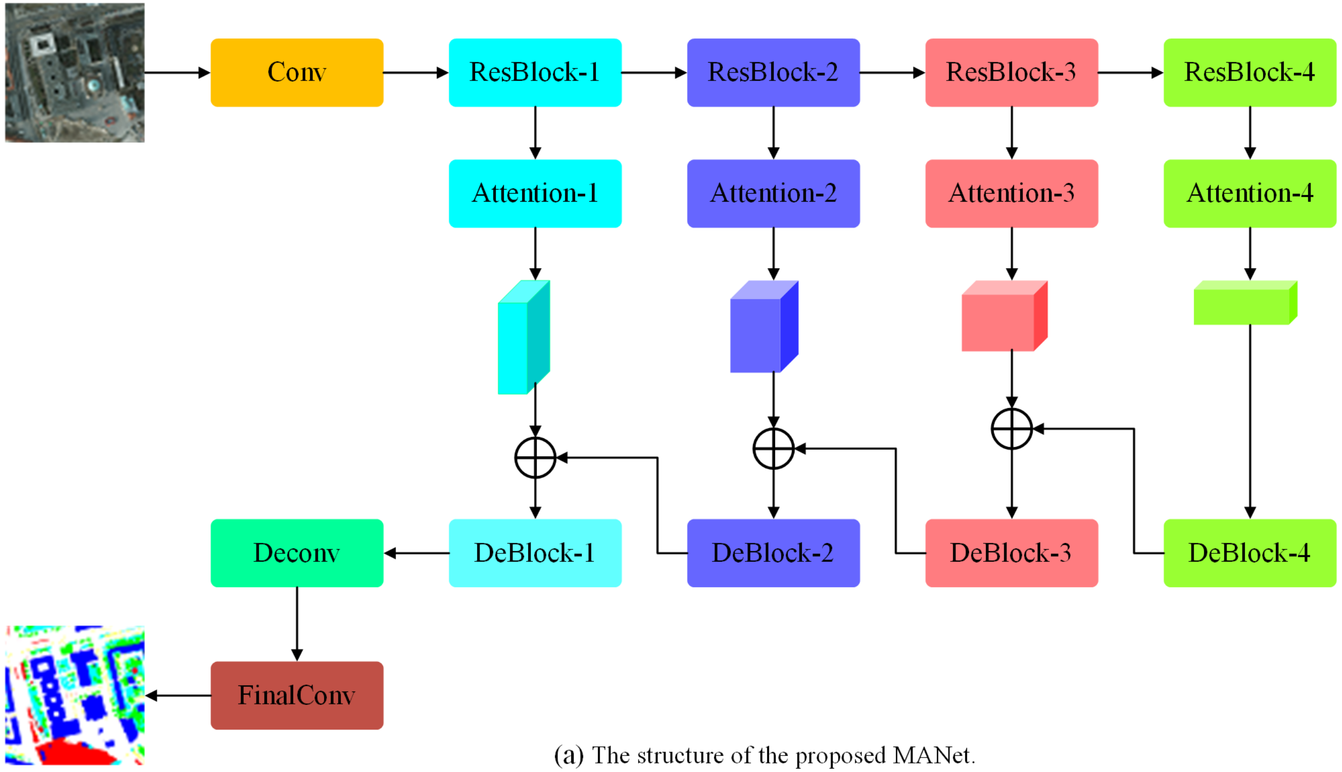
(a)



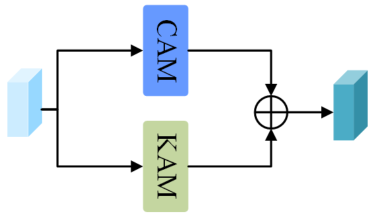
(b)

$$D(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \frac{\phi(\boldsymbol{Q})\varphi(\boldsymbol{K})^T\boldsymbol{V}}{\phi(\boldsymbol{Q})\sum_j \varphi(\boldsymbol{K})_{i,j}^T} \qquad \phi(\cdot) = \varphi(\cdot) = \text{softplus}(\cdot) \qquad \text{softplus}(x) = \log(1 + e^x)$$

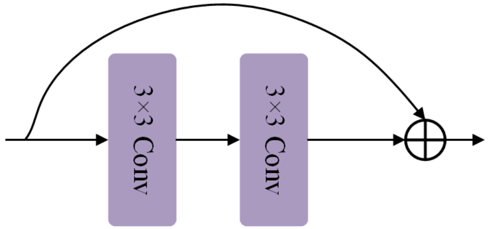
6 Results



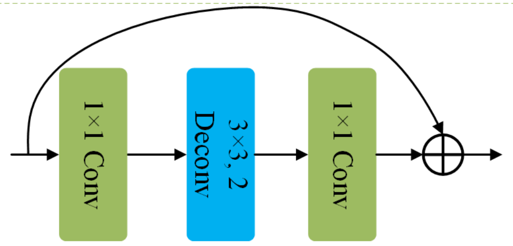
(a) The structure of the proposed MANet.



(b) The structure of the Attention Block.



(c) The structure of the ResBlock.



(d) The structure of the DeBlock.

网络结构

论文

代码

6 Results

Vaihingen

QUANTITATIVE COMPARISON RESULTS ON THE VAIHINGEN TEST SET

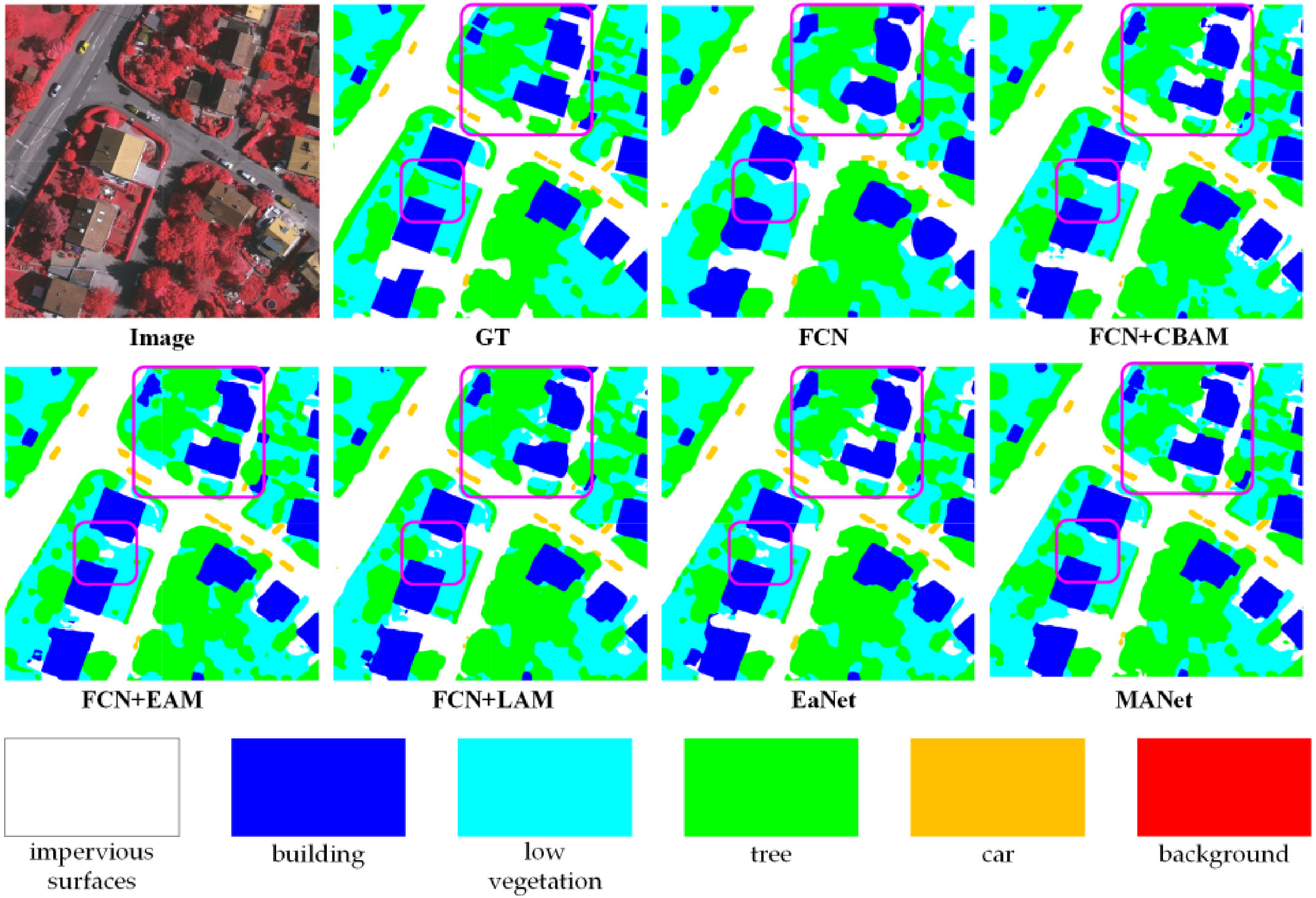
Method	Imp. surf.	Building	Low veg.	Tree	Car	Mean F1	OA (%)	mIoU (%)
FCN	89.731	93.169	80.569	88.890	71.552	84.782	87.987	73.454
FCN+SE [58]	91.886	94.604	83.185	89.379	77.084	87.228	89.711	77.894
FCN+CBAM [59]	91.592	94.766	84.195	89.494	80.877	88.185	89.956	79.612
FCN+EAM [49]	92.450	95.075	83.743	89.479	86.231	89.396	90.324	80.747
FCN+FAM [48]	92.605	94.214	84.154	**90.138**	84.897	89.202	90.304	80.664
FCN+LAM [25]	92.075	94.820	83.420	89.730	83.626	88.734	90.047	80.505
DANet [44]	91.384	94.100	83.086	89.015	76.794	86.876	89.473	77.318
PSPNet [33]	91.383	94.196	83.050	88.713	75.021	86.473	89.358	76.784
DeepLabV3+ [31]	91.630	94.086	82.505	87.991	77.656	86.774	89.124	77.127
DDCM-Net [82]	92.700	95.300	83.300	89.400	88.300	89.800	90.400	-
V-FuseNet [83]	91.000	94.400	84.500	89.900	86.300	89.200	90.000	-
LANet [84]	92.410	94.900	82.890	88.920	81.310	88.090	89.830	-
TreeUNet [85]	92.500	94.900	83.600	89.600	85.900	89.300	90.400	-
EaNet [86]	91.711	94.857	84.228	90.060	82.036	88.578	90.252	79.825
BotNet [87]	92.220	94.482	83.968	89.573	82.927	88.634	90.155	79.885
ResT [88]	92.464	95.160	83.716	89.510	84.273	89.025	90.328	80.515
Proposed MANet	**93.024**	**95.471**	**84.637**	89.978	**88.945**	**90.411**	**90.963**	**82.706**

论文

代码

6 Results

Vaihingen



论文

代码

6 Results

Potsdam

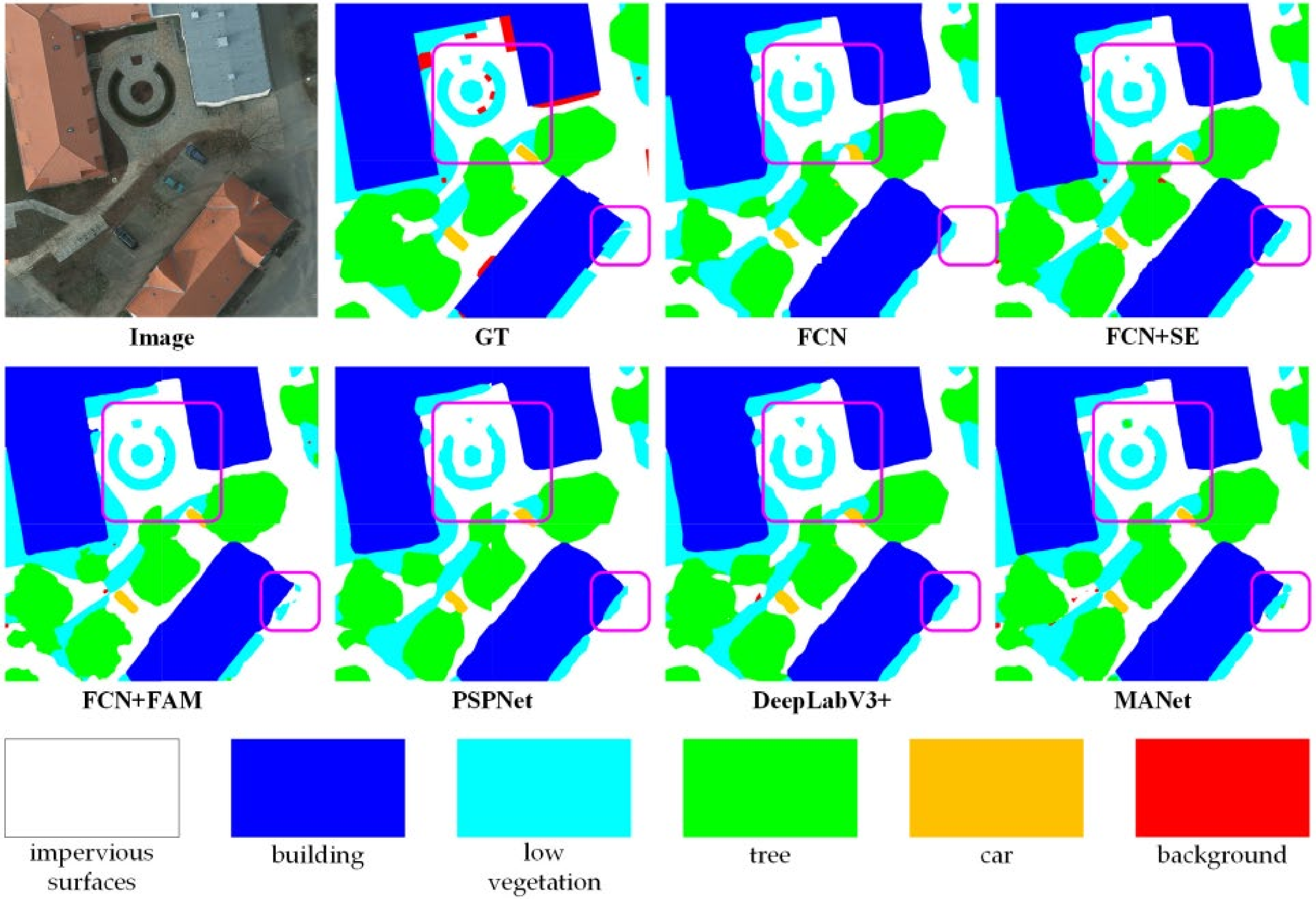
QUANTITATIVE COMPARISON RESULTS ON THE POTSDAM TEST SET

Method	Imp. surf.	Building	Low veg.	Tree	Car	Mean F1	OA (%)	mIoU (%)
FCN	90.839	95.591	84.097	84.750	84.952	88.046	88.022	79.532
FCN+SE [58]	91.647	96.118	86.078	88.009	95.077	91.386	89.598	85.380
FCN+CBAM [59]	92.719	96.127	85.773	88.217	95.827	91.733	89.898	85.648
FCN+EAM [49]	92.748	96.041	86.480	88.407	96.023	91.940	90.241	85.727
FCN+FAM [48]	92.580	96.127	86.787	88.165	95.792	91.890	90.179	85.417
FCN+LAM [25]	92.771	96.406	86.476	87.277	96.090	91.804	90.119	85.367
DANet [44]	91.944	96.348	86.003	87.673	86.010	89.596	89.728	81.399
PSPNet [33]	92.199	96.107	86.940	88.339	86.302	89.977	90.143	81.990
DeepLabV3+ [31]	92.093	95.282	85.549	86.537	94.813	90.855	89.176	84.235
DDCM-Net [82]	92.900	96.900	87.700	**89.400**	94.900	92.300	90.800	-
V-FuseNet [83]	92.700	96.300	87.300	88.500	95.400	92.040	90.600	-
TreeUNet [85]	93.100	**97.300**	86.600	87.100	95.800	91.980	90.700	-
LANet [84]	93.050	97.190	87.300	88.040	94.190	91.950	90.840	-
EaNet [86]	92.872	96.302	86.163	87.991	95.303	91.726	90.154	85.339
BotNet [87]	92.343	96.298	87.322	88.741	94.165	91.774	90.422	84.973
ResT [88]	91.139	95.106	86.296	87.267	94.627	90.887	89.128	83.500
Proposed MANet	**93.397**	96.959	**88.319**	89.360	**96.483**	**92.904**	**91.318**	**86.952**

论文

代码

6 Results



Potsdam

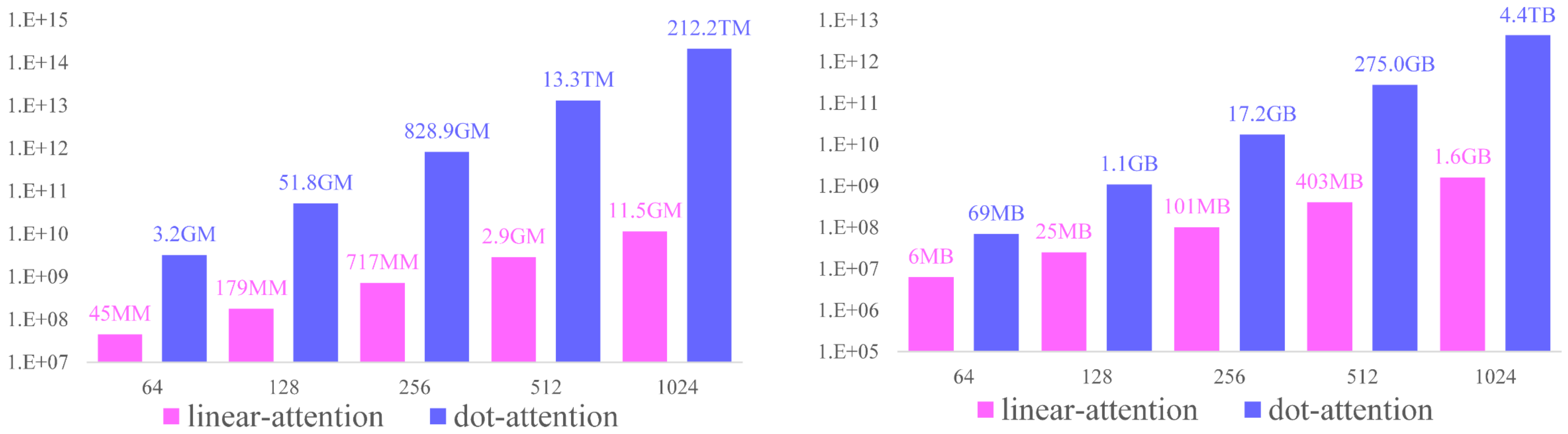
论文

代码

6 Results

方案2

泰勒展开

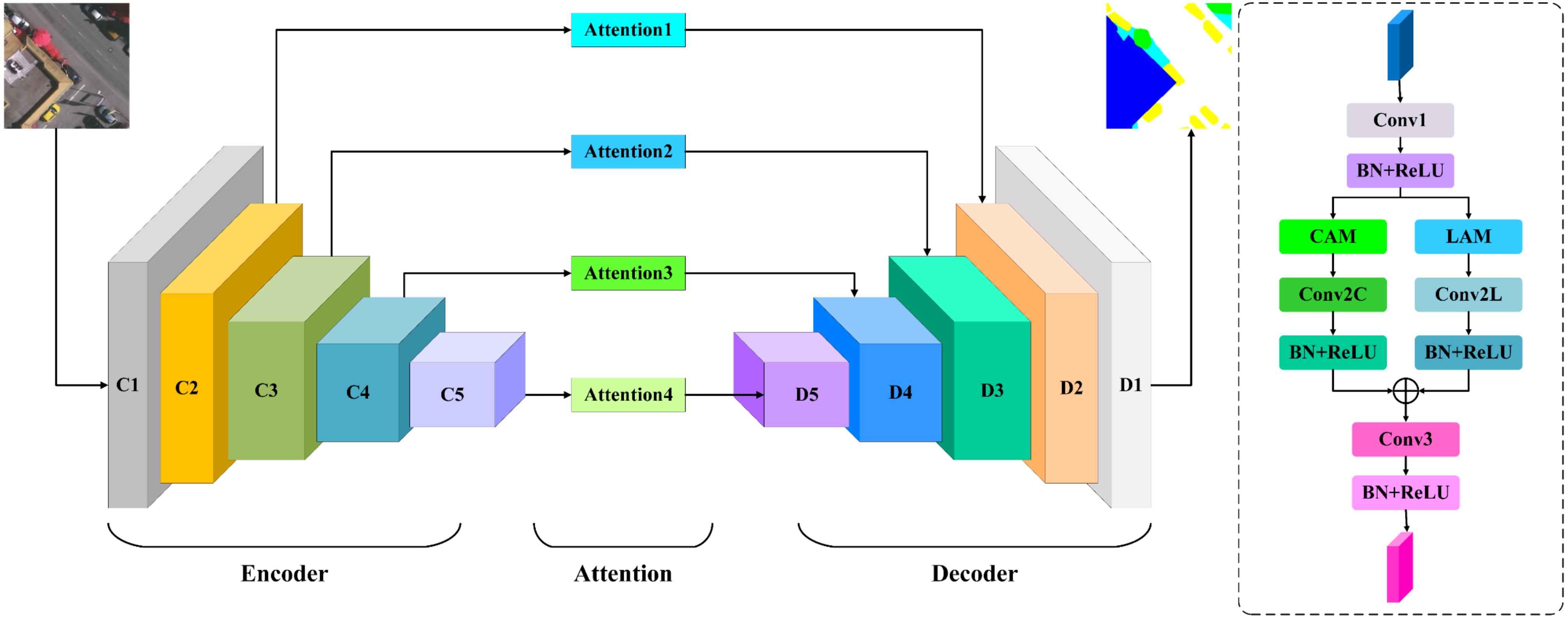


(a) (b)

$$D(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \frac{\sum_j \boldsymbol{V}_{i,j} + \left(\frac{\boldsymbol{Q}}{\|\boldsymbol{Q}\|_2}\right)\left(\left(\frac{\boldsymbol{K}}{\|\boldsymbol{K}\|_2}\right)^T \boldsymbol{V}\right)}{N + \left(\frac{\boldsymbol{Q}}{\|\boldsymbol{Q}\|_2}\right)\sum_j \left(\frac{\boldsymbol{K}}{\|\boldsymbol{K}\|_2}\right)^T_{i,j}}$$

6 Results

网络结构



(a) The Integrated Structure of Proposed MAResU-Net

(b) The Attention Block

论文

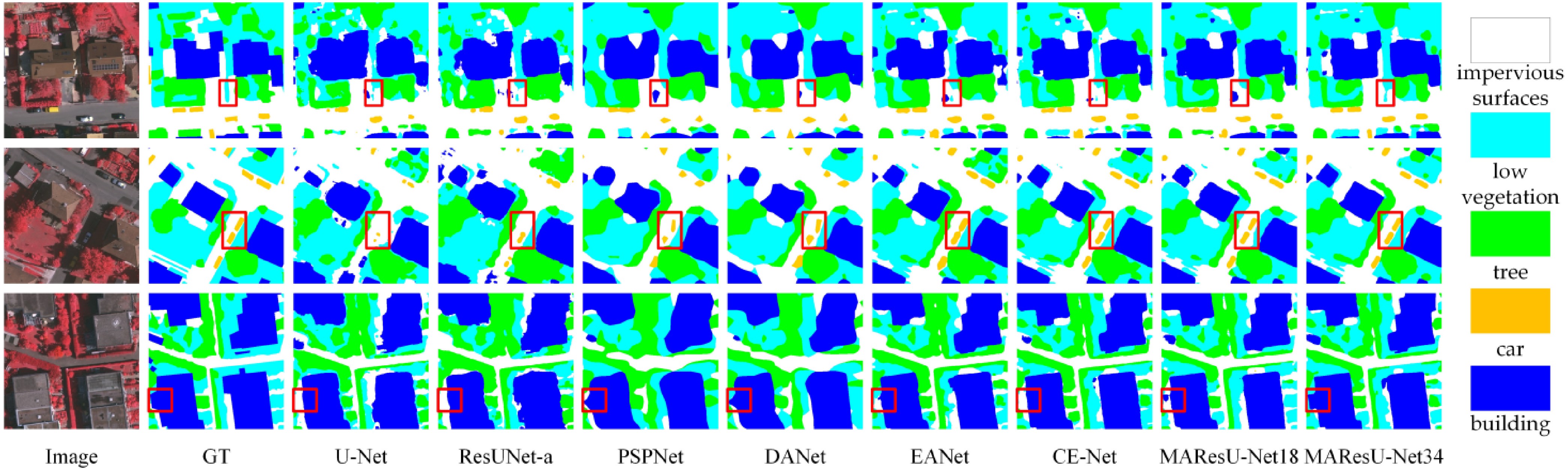
代码

6 Results

Vaihingen

EXPERIMENTAL RESULTS ON THE VAIHINGEN DATA SETS

Method	Backbone	Imp. surf.	Building	Low veg.	Tree	Car	Mean F1	OA (%)	mIoU (%)
U-Net [1]	-	84.331	86.479	73.132	83.886	40.825	73.731	82.023	61.362
ResUNet-a [2]	-	86.708	88.319	76.791	85.429	57.094	78.868	84.350	66.995
PSPNet [3]	ResNet34	90.273	94.218	82.757	88.606	51.100	81.391	88.820	71.591
DANet [5]	ResNet34	91.135	94.818	83.467	88.923	62.979	84.264	89.524	74.728
EaNet [8]	ResNet34	92.172	95.197	82.811	89.254	80.563	87.999	89.995	80.223
CE-Net [10]	ResNet34	92.681	**95.529**	83.359	89.492	81.243	88.461	90.402	81.492
MAResU-Net	ResNet18	91.971	95.044	83.735	89.349	78.283	87.676	90.047	80.749
MAResU-Net	ResNet34	**92.912**	95.256	**84.947**	**89.939**	**88.330**	**90.277**	**90.860**	**83.301**

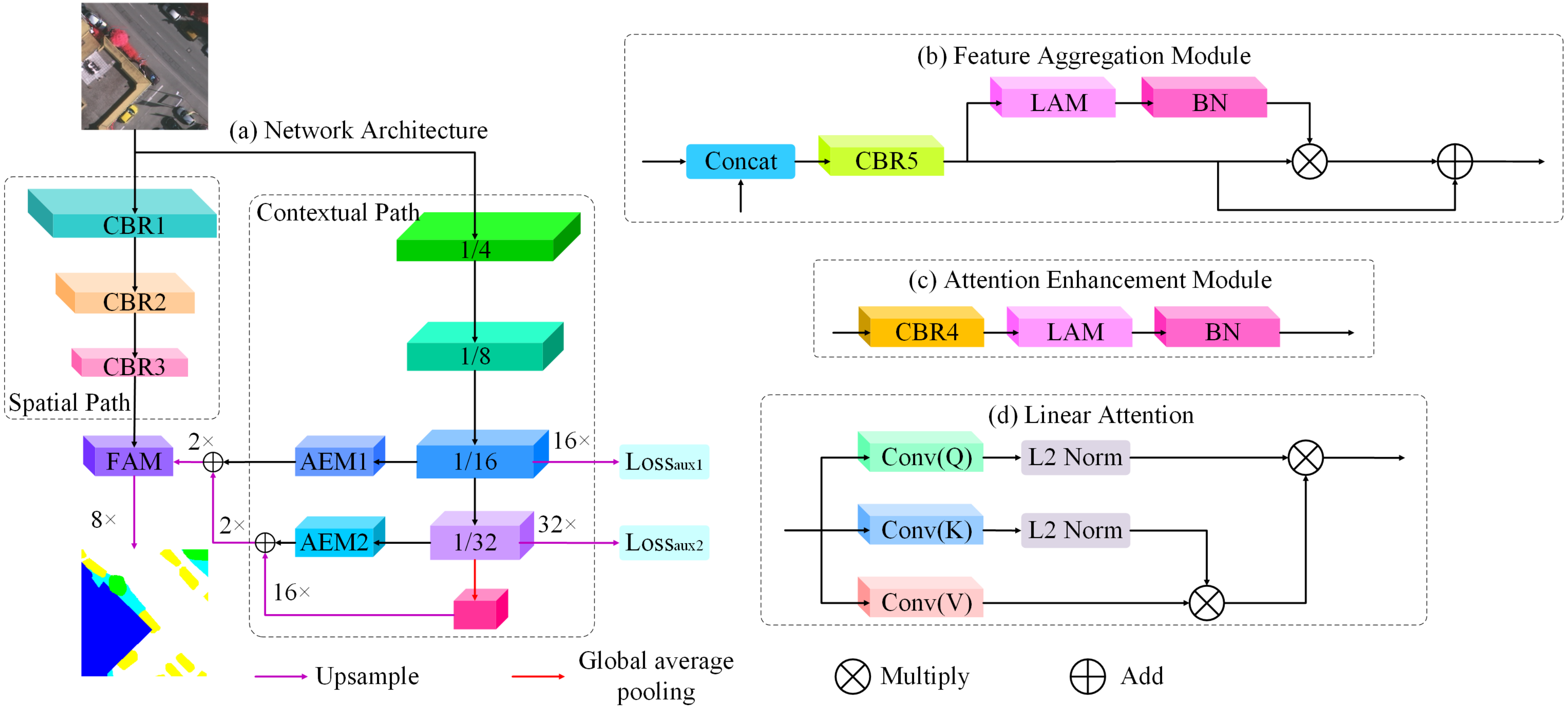


论文

代码

6 Results

网络结构



论文

代码

6 Results

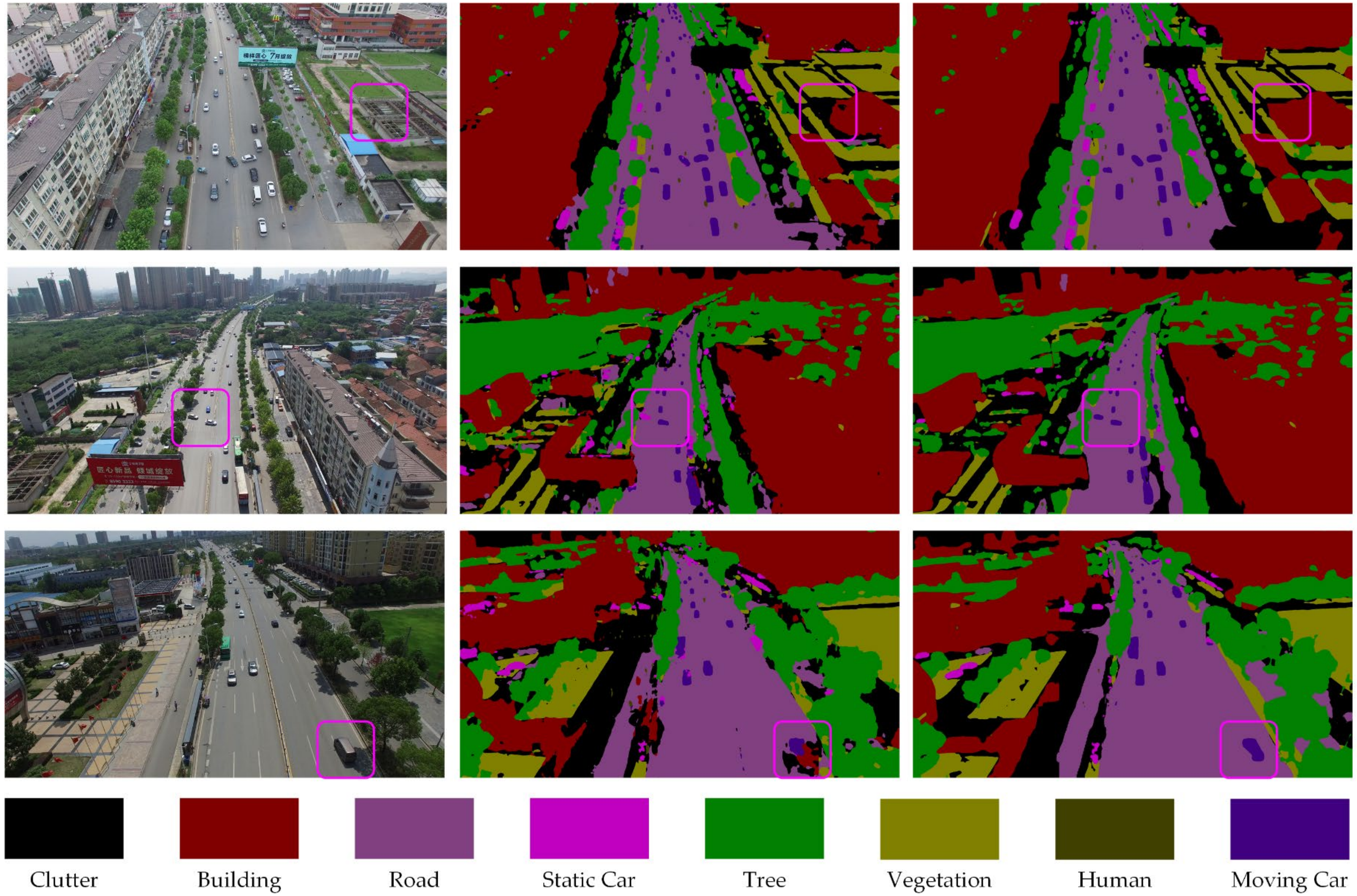
UAVid

Method	building	tree	clutter	road	vegetation	static car	moving car	human	mIoU
MSD	79.8	74.5	57.0	74.0	55.9	32.1	62.9	**19.7**	57.0
Fast-SCNN	75.7	71.5	44.2	61.6	43.4	19.5	51.6	0.0	45.9
BiSeNet	85.7	78.3	64.7	61.1	**77.3**	**63.4**	48.6	17.5	61.5
SwiftNet	85.3	78.2	64.1	61.5	76.4	62.1	51.1	15.7	61.1
ShelfNet	76.9	73.2	44.1	61.4	43.4	21.0	52.6	3.6	47.0
ABCNet	**86.4**	**79.9**	**67.4**	**81.2**	63.1	48.4	**69.8**	13.9	**63.8**

论文

代码

6 Results



UAVid

论文

代码


THANKS