# Gated Convolutional Neural Network for Semantic Segmentation in High-Resolution Images

Hongzhen Wang, Ying Wang, Qian Zhang, Shiming Xiang, Chunhong Pan

Universidade Federal de Minas Gerais
Departamento de Ciência da Computação

10 de novembro de 2017

DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

UFMG

# Introduction

- High resolution images contains a lot of complex objects with various sizes

1

[1] 2014 IEEE GRSS Data Fusion Contest.
Online:http://www.grss-ieee.org/community/technicalcommittees/data-fusion/

# Introduction

- Many objects in this image have high intra-class variance and low inter-class variance. (Grey roofs and roads, for example)

2

[2] 2014 IEEE GRSS Data Fusion Contest.
Online:http://www.grss-ieee.org/community/technicalcommittees/data-fusion/

# Introduction


- Features at different levels need to be extracted and jointly combined to fulfill the segmentation task
- High level and abstract features are more suitable for large and confused objects
- While small objects benefit from low-level and raw features
- In traditional way, we only use high-level features and the low-level feature maps are discarded

# Introduction

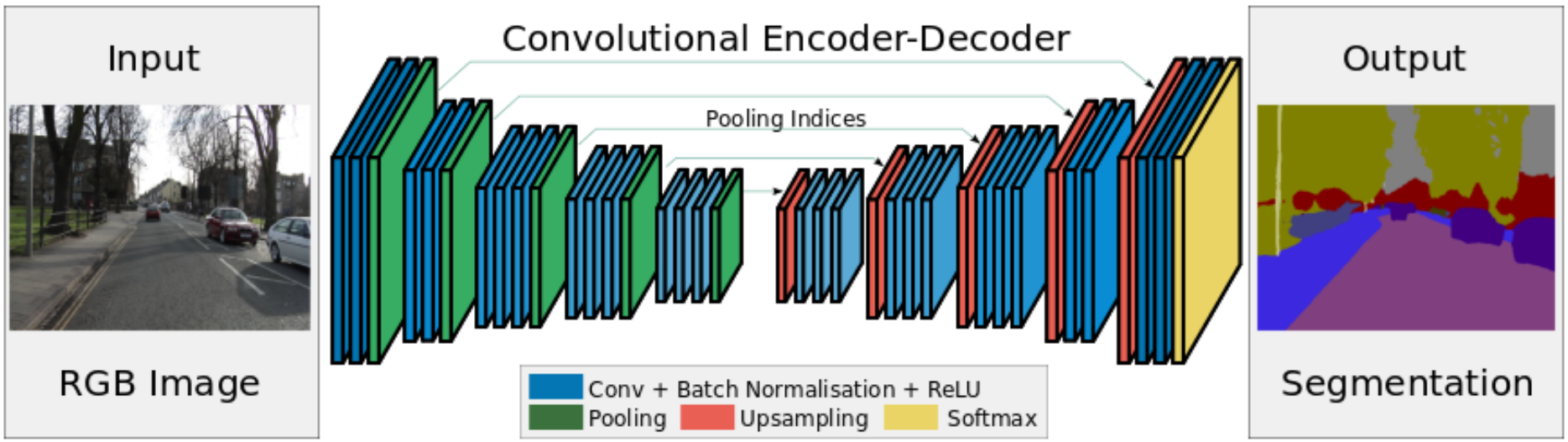



**Figura :** Segnet

[3].

[3] Vijay Badrinarayanan, Alex Kendall e Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation". Em: *arXiv preprint arXiv:1511.00561* (2015).

# Introduction

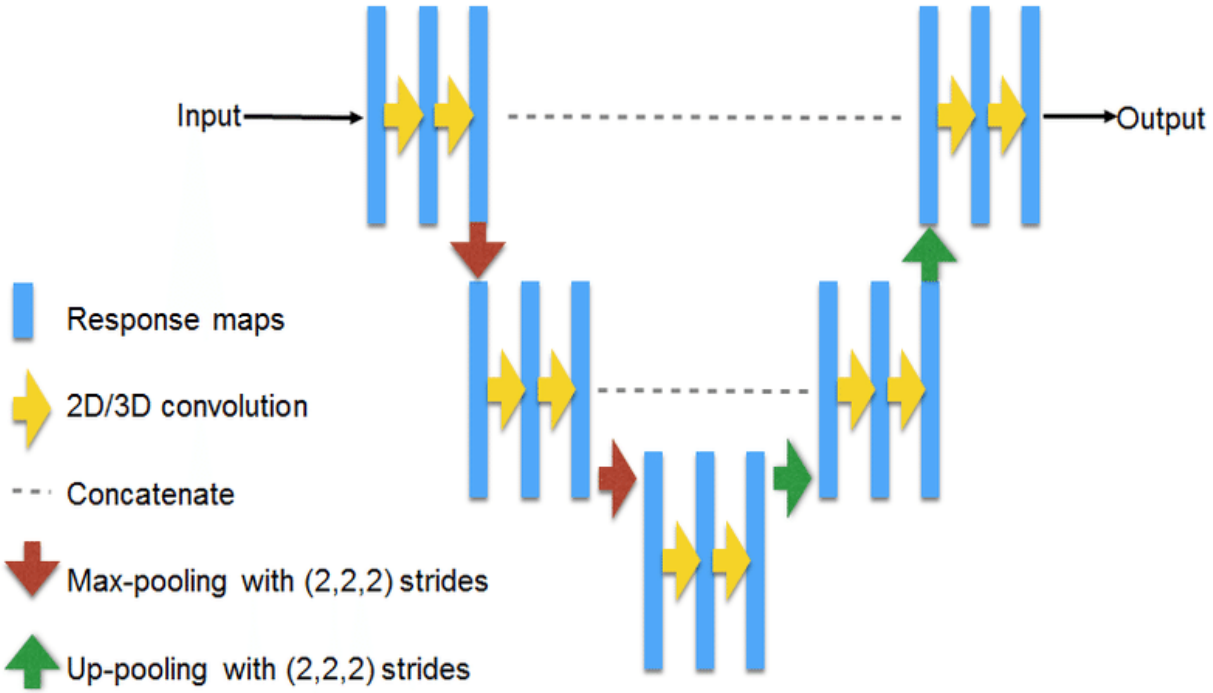



**Figura :** UNet

[4] Olaf Ronneberger, Philipp Fischer e Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". Em: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2015, pp. 234–241.

# Introduction


Without feature selection:

- Redundant information can result in oversegmentation when the model tends to receive more information from lower layers
- Fine-grained details can be lose and lead to under-segmentation when the networks tens to receive more information from upperlayers

# Introduction

## Hypothesis


- Is using all the features really the best alternative?

# Introduction

## Hypothesis


- Is using all the features really the best alternative?
  - If no. How to make a selection?

# Introduction

## Entropy


- Generally, entropy refers to disorder or uncertainty

$$H(x) = E[-log_2(p_i(x))] = -\sum_{i=1}^{k} p_i(x) log_2(p_i(x)) \quad (1)$$

where:

$E[.]$ denotes expectation over all the k categories
$p_i(x)$ is the probability of pixel x belonging to category i

# Introduction

## Entropy


P(x) = [0.8,0.1,0.1] $\quad log_2 0.8 = -0.3219 \quad log_2 0.1 = -3.3219$

# Introduction

## Entropy


P(x) = [0.8,0.1,0.1] $\quad log_2 0.8 = -0.3219 \quad log_2 0.1 = -3.3219$

$$H(x) = E[-log_2(p_i(x))] = -\sum_{i=1}^{k} p_i(x) log_2(p_i(x))$$

# Introduction

## Entropy


P(x) = [0.8,0.1,0.1] $\quad log_2 0.8 = -0.3219 \quad log_2 0.1 = -3.3219$

$$H(x) = E[-log_2(p_i(x))] = -\sum_{i=1}^{k} p_i(x) log_2(p_i(x))$$

$$H(x) = -[(0.8x(-0.3219)) + (0.1x(-3.3219)) + (0.1x(-3.3219))]$$

# Introduction

## Entropy


P(x) = [0.8,0.1,0.1] $\quad log_2 0.8 = -0.3219 \quad log_2 0.1 = -3.3219$

$$H(x) = E[-log_2(p_i(x))] = -\sum_{i=1}^{k} p_i(x) log_2(p_i(x))$$

$$H(x) = -[(0.8x(-0.3219)) + (0.1x(-3.3219)) + (0.1x(-3.3219))]$$

$$H(x) = -[-0.257522 + (-0.33219) + (-0.33219)]$$

# Introduction

## Entropy


P(x) = [0.8,0.1,0.1] $\quad log_2 0.8 = -0.3219 \quad log_2 0.1 = -3.3219$

$$H(x) = E[-log_2(p_i(x))] = -\sum_{i=1}^{k} p_i(x) log_2(p_i(x))$$

$$H(x) = -[(0.8x(-0.3219)) + (0.1x(-3.3219)) + (0.1x(-3.3219))]$$

$$H(x) = -[-0.257522 + (-0.33219) + (-0.33219)]$$

$$H(x) = -[-0.9219]$$

# Introduction

## Entropy

P(x) = [0.8,0.1,0.1] $\quad log_2 0.8 = -0.3219 \quad log_2 0.1 = -3.3219$

$$H(x) = E[-log_2(p_i(x))] = -\sum_{i=1}^{k} p_i(x) log_2(p_i(x))$$

$$H(x) = -[(0.8x(-0.3219)) + (0.1x(-3.3219)) + (0.1x(-3.3219))]$$

$$H(x) = -[-0.257522 + (-0.33219) + (-0.33219)]$$

$$H(x) = -[-0.9219]$$

$$H(x) = 0.9219$$

# Introduction

## Entropy

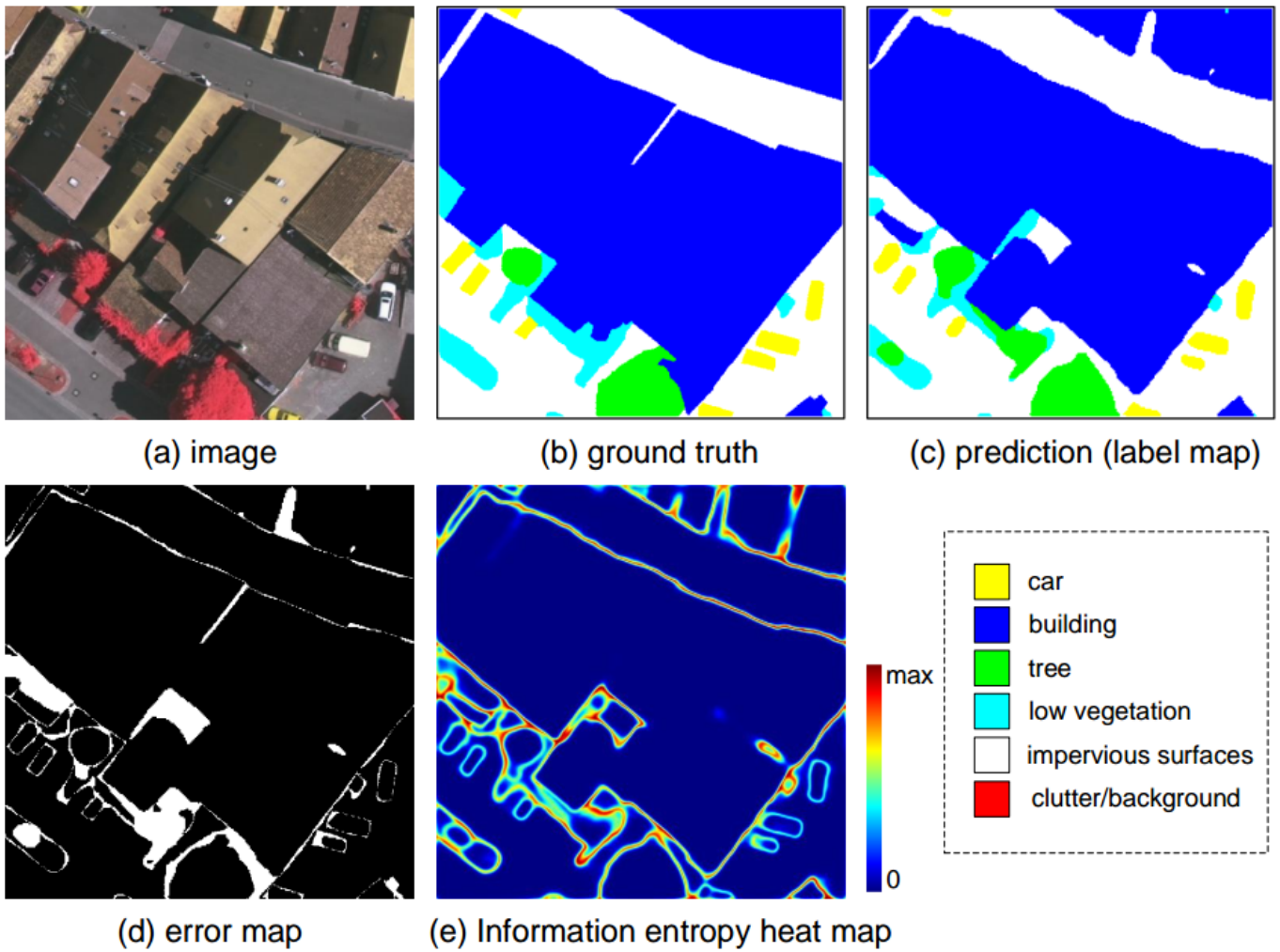

P(x) = [0.8,0.1,0.1] $H(x) = 0.9219$

P(x) = [0.5,0.4,0.1] $H(x) = 1,3601$

P(x) = [0.3,0.3,0.4] $H(x) = 1,5709$

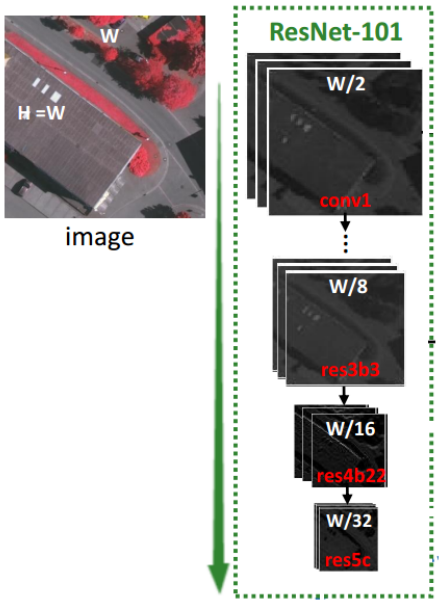
When the entropy of pixel x is maximized p(x) approximates an uniform probability distribution. In this case, the network is unable to classify this pixel using only existing information

# Introduction



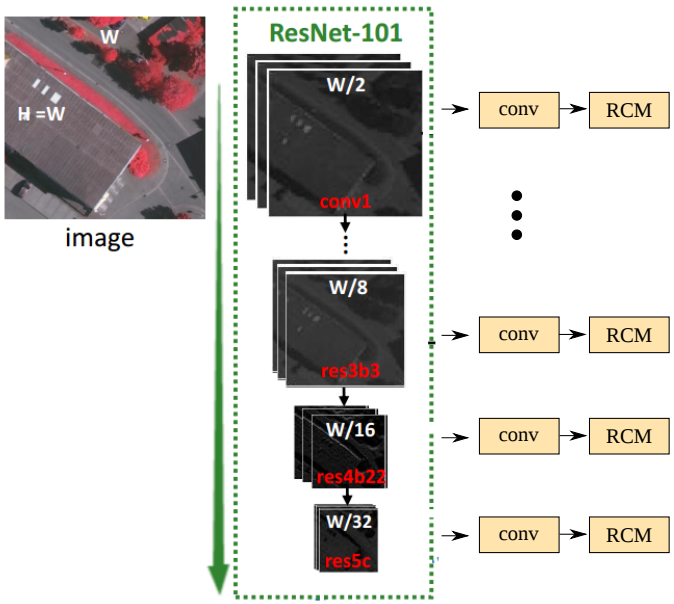
# Approach

- Features are extracted in a pretrained ResNet



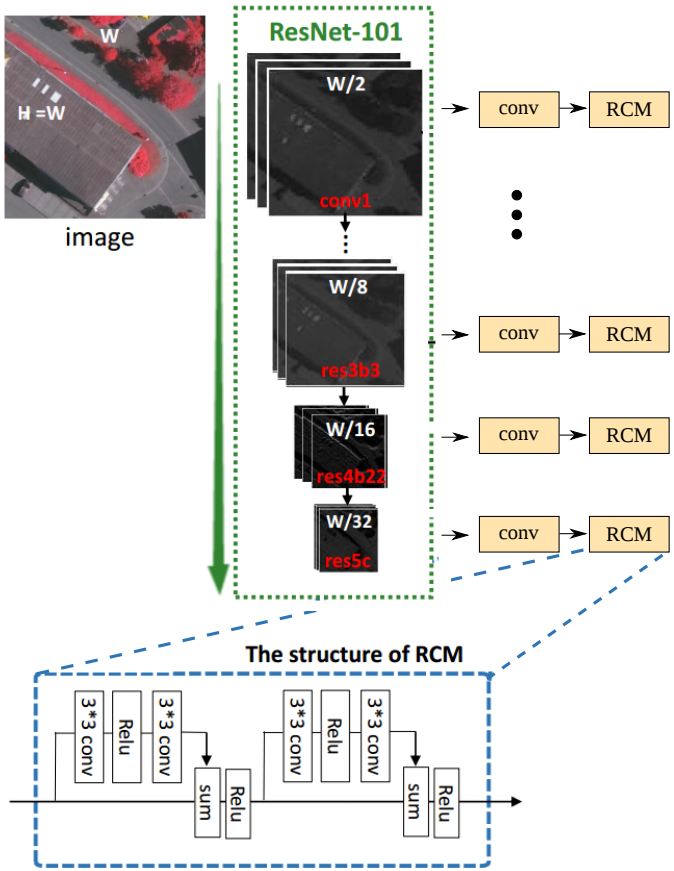
# Approach

- Each set of features is submitted to a convolutional layer followed by a RCM module



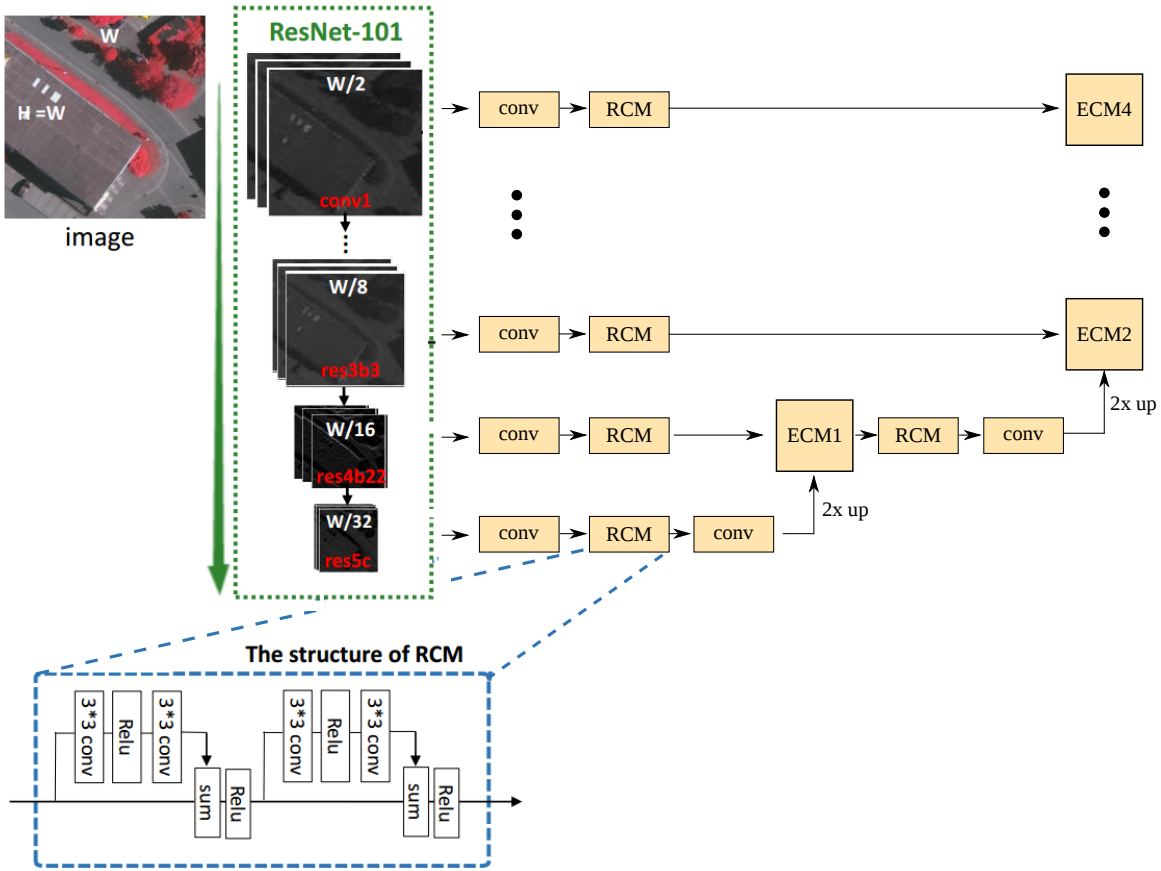
# Approach

- RCM module are based in residual blocks in ResNet and is used to ease the training and avoid the gradient vanishing problem



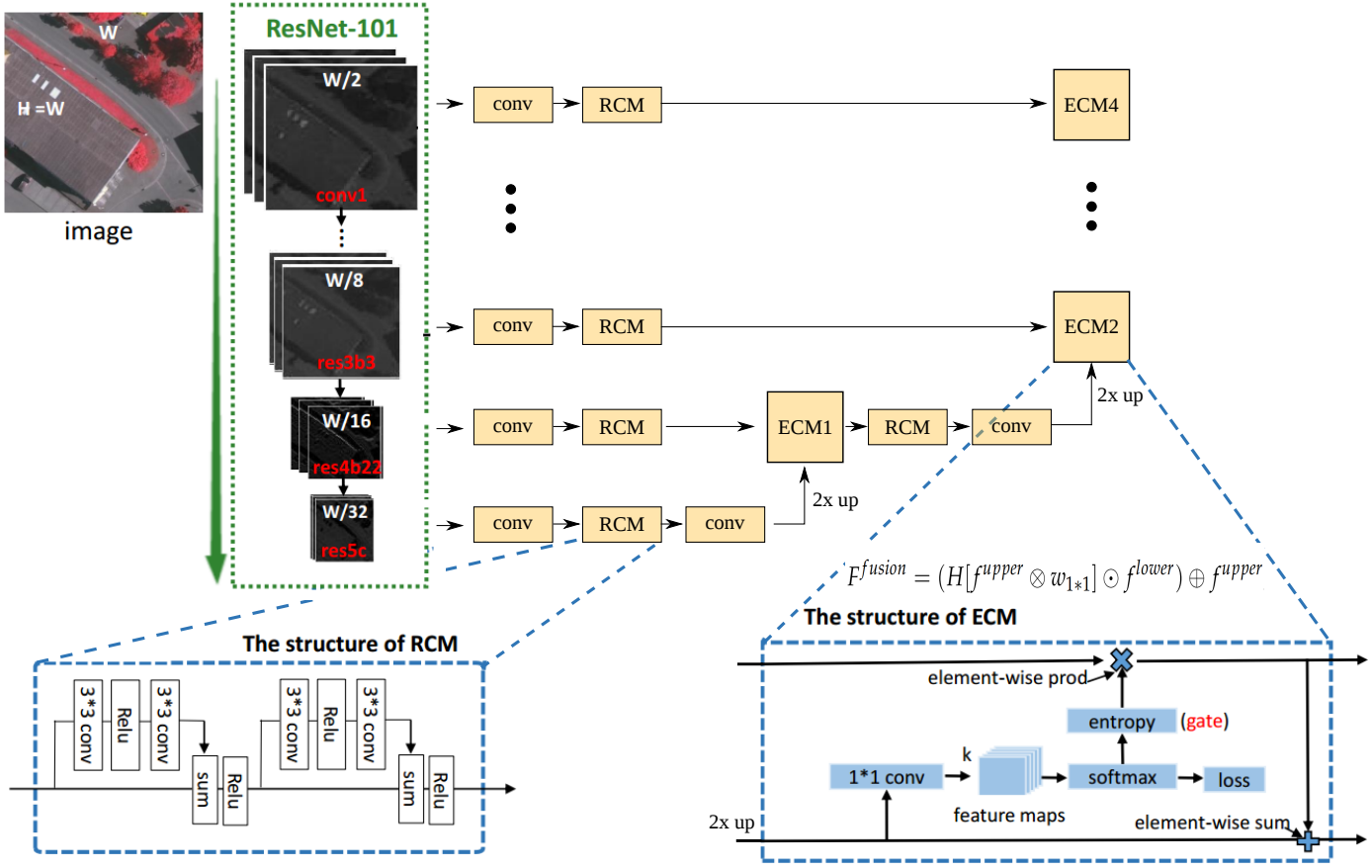
# Approach

- Each pair of output of RCM module is used as input for the ECM module starting from the higher layer until the low layer



# Approach

- The ECM module is used to fuse higher feature maps and low feature maps



# Approach

$$F^{fusion} = (H[f^{upper} \otimes w_{1*1}] \odot f^{lower}) \oplus f^{upper}$$

# Approach

$$F^{fusion} = \left(H[f^{upper} \otimes w_{1*1}] \odot f^{lower}\right) \oplus f^{upper}$$

$$\begin{bmatrix} 0.9219 & 1.3601 \\ 0.9219 & 1.5709 \end{bmatrix} \qquad \begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix} \qquad \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}$$

# Approach

$$F^{fusion} = (H[f^{upper} \otimes w_{1*1}] \odot f^{lower}) \oplus f^{upper}$$

$$F^{fusion} = \begin{bmatrix} 0.9219 & 1.3601 \\ 0.9219 & 1.5709 \end{bmatrix} \odot \begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix} \oplus \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}$$

element-wise product operator

element-wise sum operator

# Approach

$$F^{fusion} = (H[f^{upper} \otimes w_{1*1}] \odot f^{lower}) \oplus f^{upper}$$

$$F^{fusion} = \begin{bmatrix} 0.9219 & 1.3601 \\ 0.9219 & 1.5709 \end{bmatrix} \odot \begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix} \oplus \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}$$

element-wise product operator

element-wise sum operator

$$F^{fusion} = \begin{bmatrix} 0.1843 \end{bmatrix}$$

# Approach

$$F^{fusion} = (H[f^{upper} \otimes w_{1*1}] \odot f^{lower}) \oplus f^{upper}$$

$$F^{fusion} = \begin{bmatrix} 0.9219 & 1.3601 \\ 0.9219 & 1.5709 \end{bmatrix} \odot \begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix} \oplus \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}$$

element-wise product operator

element-wise sum operator

$$F^{fusion} = \begin{bmatrix} 0.1843 & 1.0880 \end{bmatrix}$$

# Approach

$$F^{fusion} = (H[f^{upper} \otimes w_{1*1}] \odot f^{lower}) \oplus f^{upper}$$

$$F^{fusion} = \begin{bmatrix} 0.9219 & 1.3601 \\ 0.9219 & 1.5709 \end{bmatrix} \odot \begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix} \oplus \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}$$

element-wise product operator

element-wise sum operator

$$F^{fusion} = \begin{bmatrix} 0.1843 & 1.0880 \\ 0.7375 & 0.3141 \end{bmatrix} \oplus \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix} = \begin{bmatrix} 0.5843 & 1.6880 \\ 1.3375 & 0.7141 \end{bmatrix}$$

# Approach

$$F^{fusion} = (H[f^{upper} \otimes w_{1*1}] \odot f^{lower}) \oplus f^{upper}$$

$$F^{fusion} = \begin{bmatrix} 0.9219 & 1.3601 \\ 0.9219 & 1.5709 \end{bmatrix} \odot \begin{bmatrix} 0.2 & 0.8 \\ 0.8 & 0.2 \end{bmatrix} \oplus \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}$$

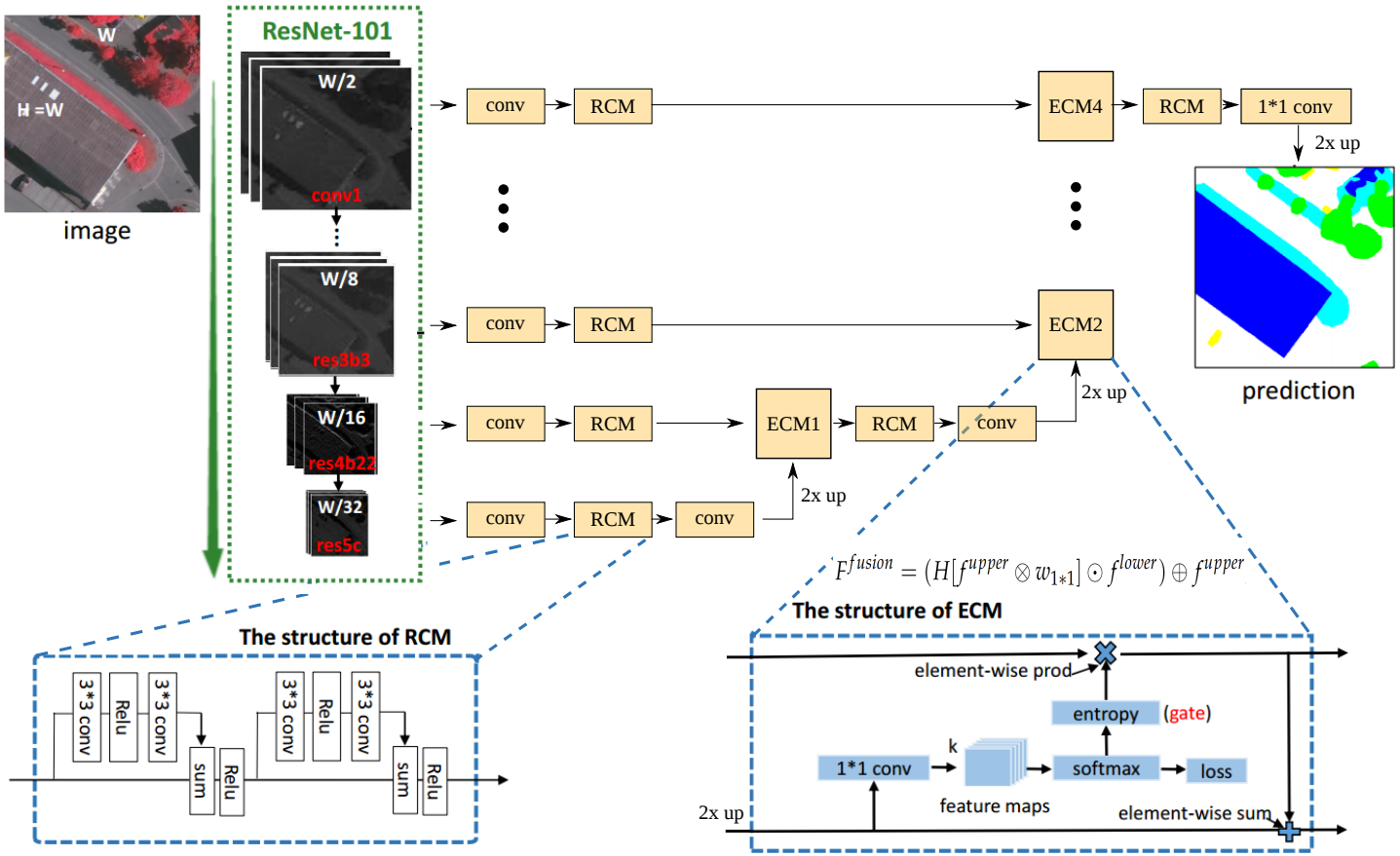
element-wise product operator

element-wise sum operator

$$F^{fusion} = \begin{bmatrix} 0.1843 & 1.0880 \\ 0.7375 & 0.3141 \end{bmatrix} \oplus \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix} = \begin{bmatrix} 0.5843 & 1.6880 \\ 1.3375 & 0.7141 \end{bmatrix}$$

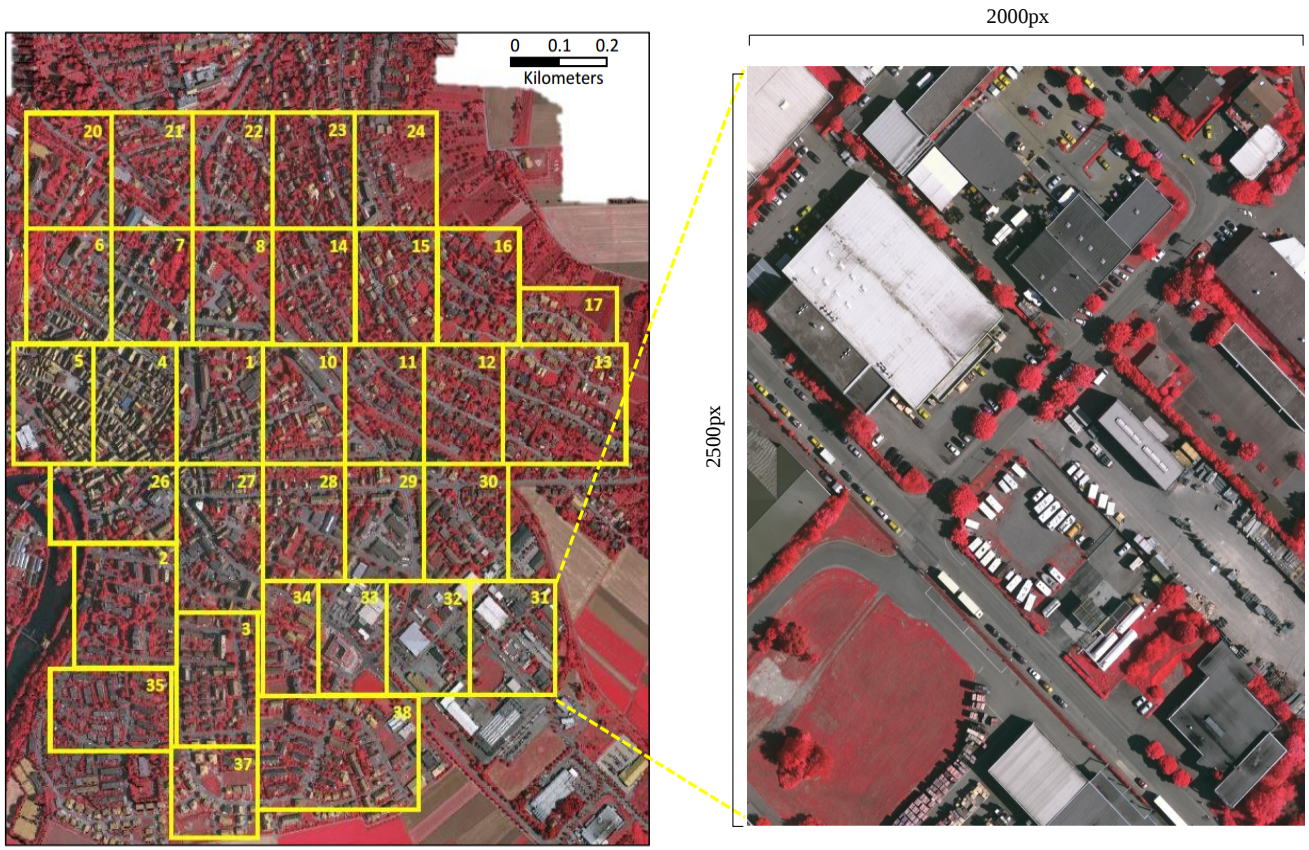
# Approach

- The final architeture is used to classify the test image



# Experimental evaluation

## Dataset




- 16 tiles are used ( 12 for train and 4 for validation)
- Each tile 2500x2000 pixels with 9 cm of resolution
- Manually classified into six classes
- Metrics: F1 score and Overall Accuracy

$$F1 = 2 \times \frac{precision \times recall}{precision + recall}$$

$$precision = \frac{TP}{TP + FP} \qquad recall = \frac{TP}{TP + FN}$$

$$Overall\ Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

TP = true positive
TN = true negative
FP = false positive
FN = false negative

Figure: Overview of the ISPRS 2D Vaihingen Labelling dataset. There are 33 tiles. Numbers in the figure refer to the individual flag.

# Experimental evaluation


- Baseline - GSN without entropy control module
- GSN\GSN_noL - GSN with\without auxiliary loss in ECM, respectively
- GSN_w - classes with different weigths
- GSN_w_ mc - with sliding window overlap and multi-scale input

| Method | Imp Surf | Building | Low_veg | Tree | Car | Overall Accuracy | Mean $F_1$ Score |
|---|---|---|---|---|---|---|---|
| baseline | 87.6% | 93.2% | 73.3% | 86.9% | 54.1% | 86.1% | 79.0% |
| GSN | 89.2% | 94.5% | 74.9% | 87.5% | 79.8% | 87.9% | 85.2% |
| GSN_noL | 89.1% | 94.3% | 74.7% | 87.4% | 78.7% | 87.8% | 84.8% |
| GSN_w | 89.5% | 94.4% | 75.9% | 87.8% | 80.9% | 88.3% | 85.7% |
| GSN_w_mc | **90.2%** | **94.8%** | **76.9%** | **88.3%** | **82.3%** | **88.9%** | **86.5%** |

# Experimental evaluation

| Method | Imp Surf | Building | Low_veg | Tree | Car | Overall Accuracy | Mean $F_1$ Score |
|---|---|---|---|---|---|---|---|
| baseline | 87.6% | 93.2% | 73.3% | 86.9% | 54.1% | 86.1% | 79.0% |
| GSN | 89.2% | 94.5% | 74.9% | 87.5% | 79.8% | 87.9% | 85.2% |
| GSN_noL | 89.1% | 94.3% | 74.7% | 87.4% | 78.7% | 87.8% | 84.8% |
| GSN_w | 89.5% | 94.4% | 75.9% | 87.8% | 80.9% | 88.3% | 85.7% |
| GSN_w_mc | **90.2%** | **94.8%** | **76.9%** | **88.3%** | **82.3%** | **88.9%** | **86.5%** |

| Method | Imp Surf | Building | Low_veg | Tree | Car | Overall Accuracy | Mean $F_1$ Score |
|---|---|---|---|---|---|---|---|
| FCN-8s [12] | 87.1% | 91.8% | **75.2%** | 86.1% | 63.8% | 85.9% | 80.8% |
| SegNet [14] | 82.7% | 89.1% | 66.3% | 83.9% | 55.7% | 82.1% | 75.5% |
| Deeplab-v2 [21] | 88.5% | 93.5% | 73.9% | 86.9% | **84.7%** | 86.9% | 83.5% |
| RefineNet [15] | 88.1% | 93.3% | 74.0% | 87.1% | 65.1% | 86.7% | 81.5% |
| GSN | **89.2%** | **94.5%** | 74.9% | **87.5%** | 79.8% | **87.9%** | **85.2%** |

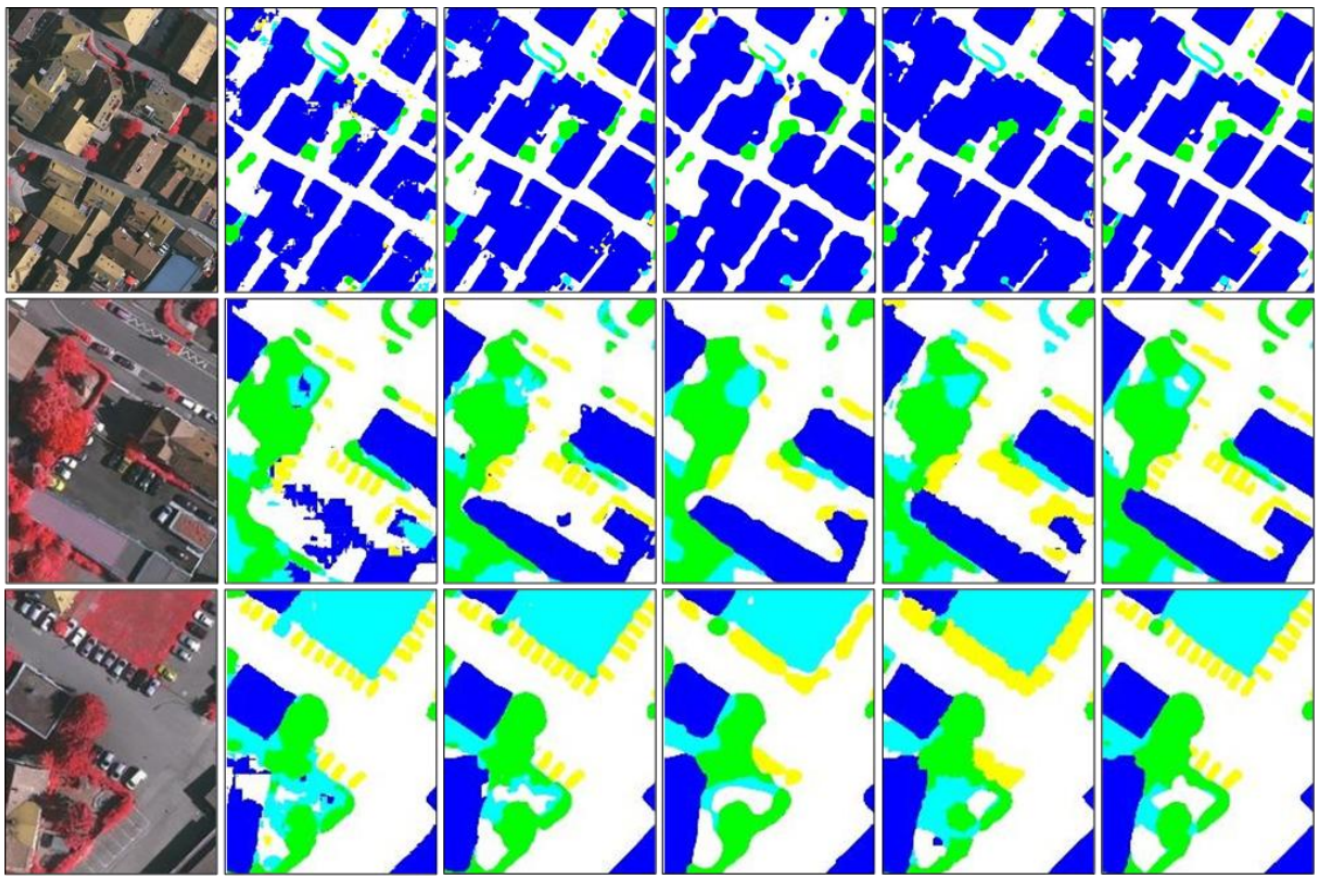
# Experimental evaluation


- Evalation of the ISPRS organizers

| Method | Imp Surf | Building | Low_veg | Tree | Car | Overall Accuracy | Mean $F_1$ Score |
|---|---|---|---|---|---|---|---|
| UPB [43] | 87.5% | 89.3% | 77.3% | 85.8% | 77.1% | 85.1% | 83.4% |
| ETH_C [44] | 87.2% | 92.0% | 77.5% | 87.1% | 54.5% | 85.9% | 79.7% |
| UOA [45] | 89.8% | 92.1% | 80.4% | 88.2% | 82.0% | 87.6% | 86.5% |
| ADL_3 [26] | 89.5% | 93.2% | 82.3% | 88.2% | 63.3% | 88.0% | 83.3% |
| RIT_2 [46] | 90.0% | 92.6% | 81.4% | 88.4% | 61.1% | 88.0% | 82.7% |
| DST_2 [8] | 90.5% | 93.7% | 83.4% | 89.2% | 72.6% | 89.1% | 85.9% |
| ONE_7 [47] | 91.0% | 94.5% | 84.4% | 89.9% | 77.8% | 89.8% | 87.5% |
| DLR_9 [28] | **92.4%** | **95.2%** | **83.9%** | **89.9%** | 81.2% | **90.3%** | 88.5% |
| GSN | 92.2% | 95.1% | 83.7% | **89.9%** | **82.4%** | **90.3%** | **88.7%** |

# Experimental evaluation

- Visual comparisons between GSN and other related methods on ISPRS test set



# Conclusions


- The ECM can effectively help for integrating contextual information from the upper layers and details from the lower layers
- The approach has the potential to perform better. Actually, the pixels in a certain region are interrelated. However, we calculate the entropy map (gate) pixel-to-pixel, which ignores the relationships between surrounding pixels.

# References I



Badrinarayanan, Vijay, Alex Kendall e Roberto Cipolla. “Segnet: A deep convolutional encoder-decoder architecture for image segmentation”. Em: *arXiv preprint arXiv:1511.00561* (2015).

He, Kaiming et al. “Deep residual learning for image recognition”. Em: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

Ronneberger, Olaf, Philipp Fischer e Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. Em: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2015, pp. 234–241.

Wang, Hongzhen et al. “Gated Convolutional Neural Network for Semantic Segmentation in High-Resolution Images”. Em: *Remote Sensing* 9.5 (2017), p. 446.