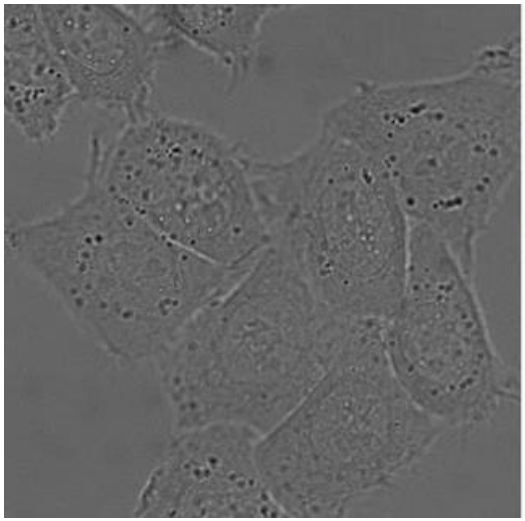
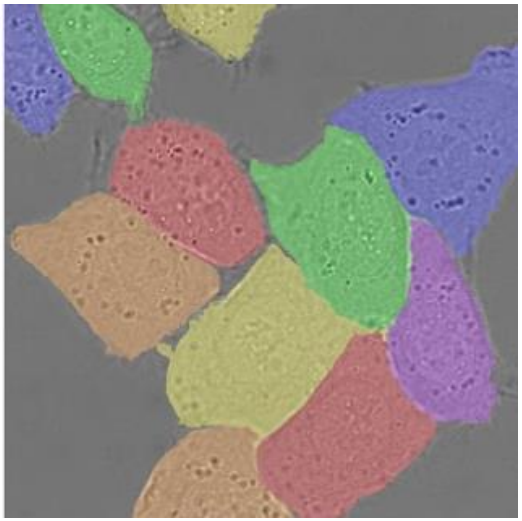
U-Net :
Convolutional Networks for Biomedical Image Segmentation

Computer Vision & Augmented Reality 연구실
학부연구생 강 준 구

Contents

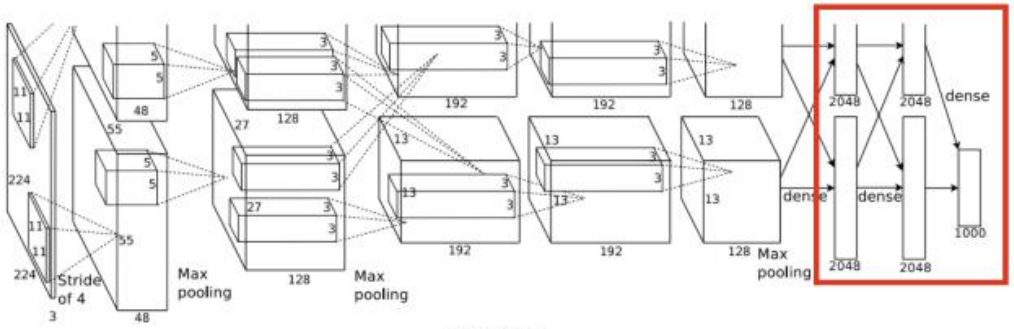
- Introduction
- Network Architecture
  - Contracting path
  - Expansive path
- Training
  - Data Augmentation
- Experiments
- Conclusion



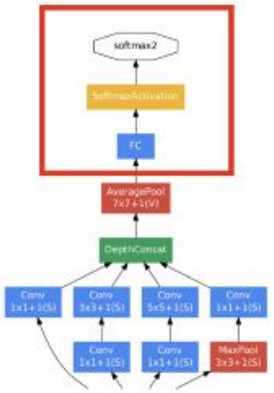


Introduction

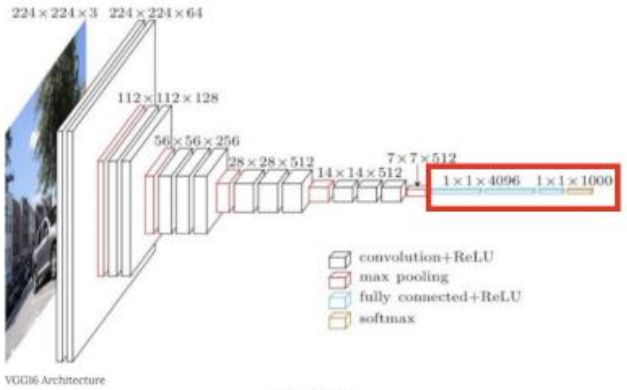
- Image Classification



AlexNet



GoogLeNet



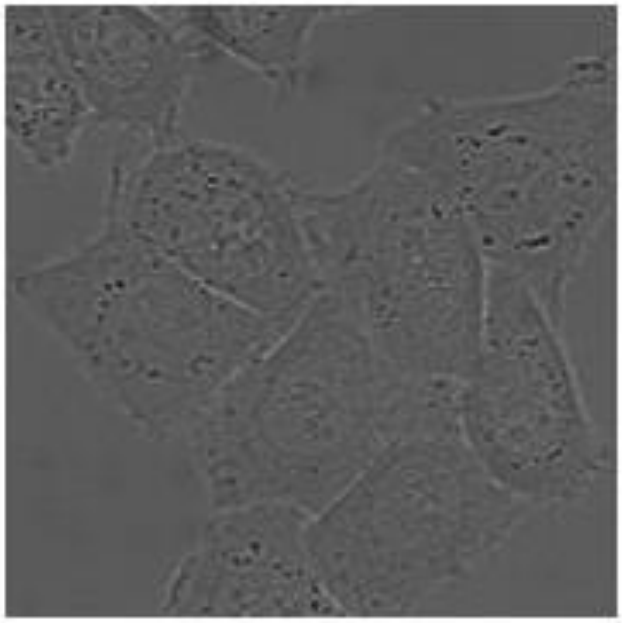
VGG16

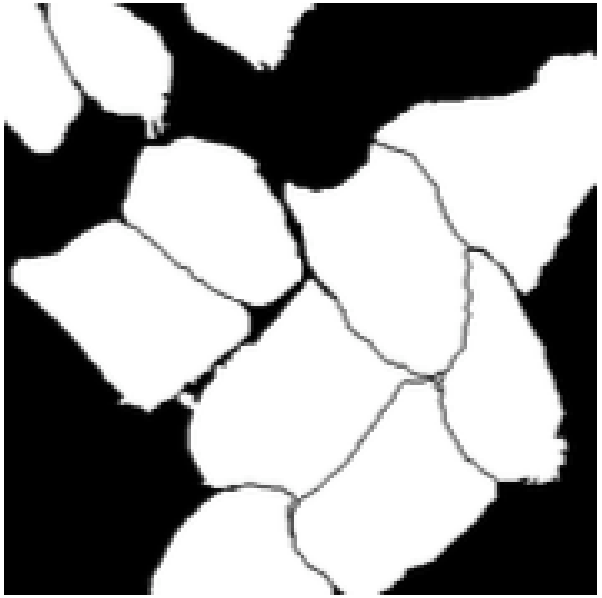
Introduction

- Semantic Segmentation

Introduction

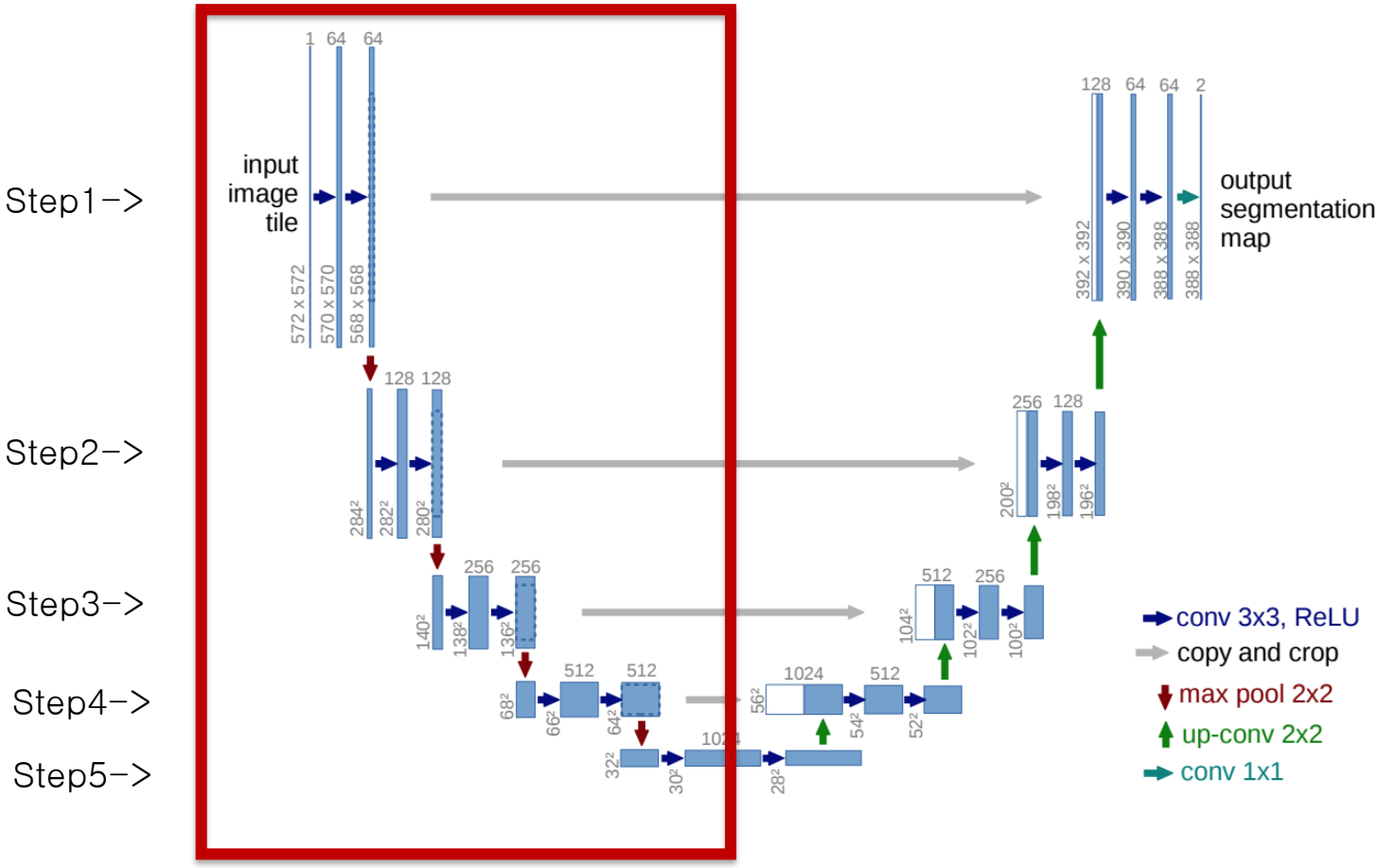
- Semantic Segmentation
  - Biomedical Image





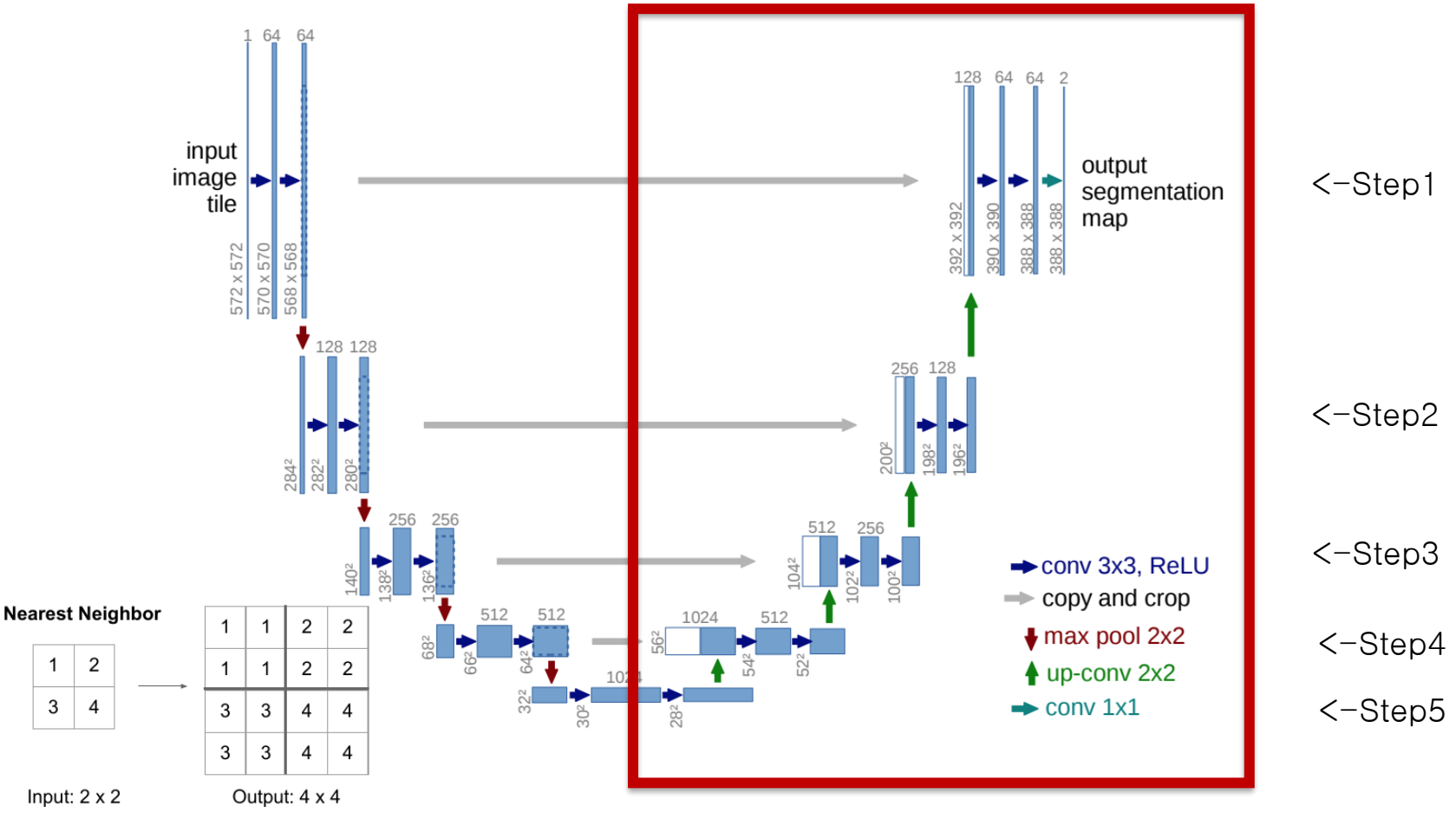
Network Architecture

- Contracting path



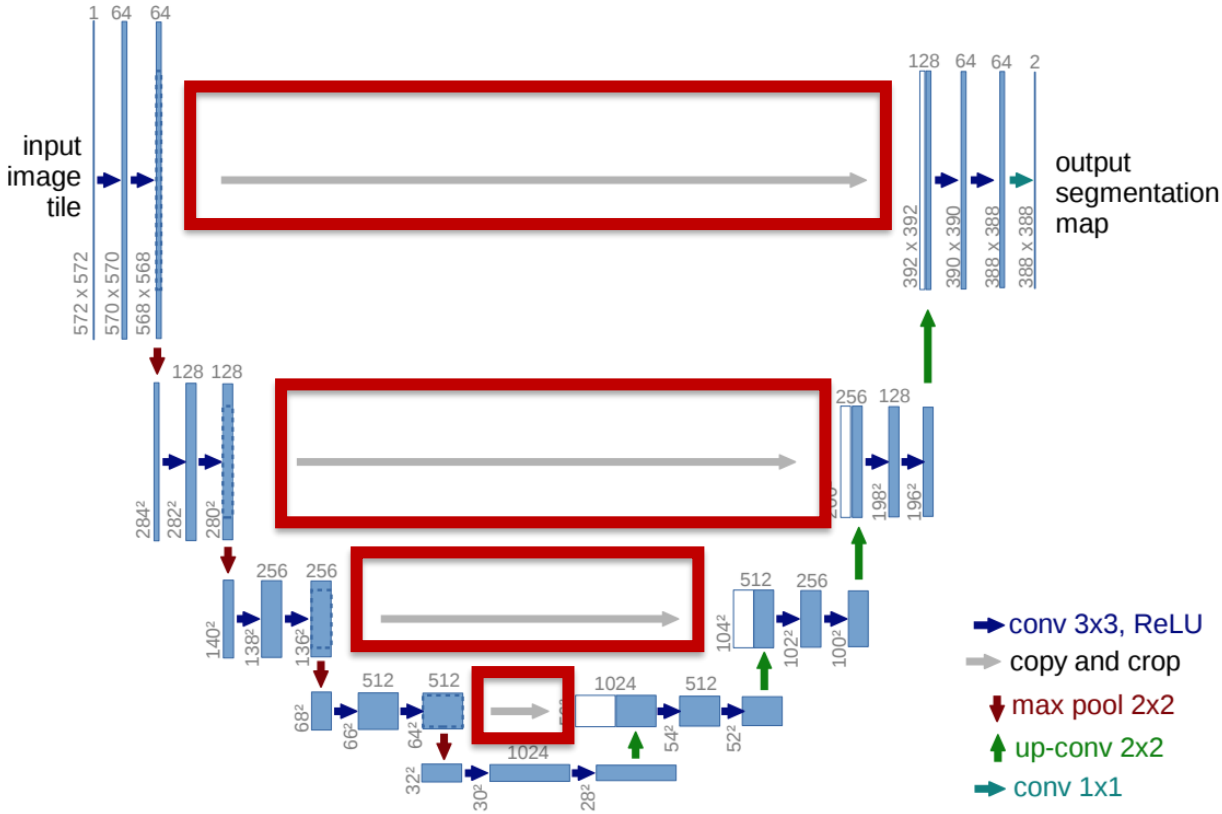
Network Architecture

- Expansive path

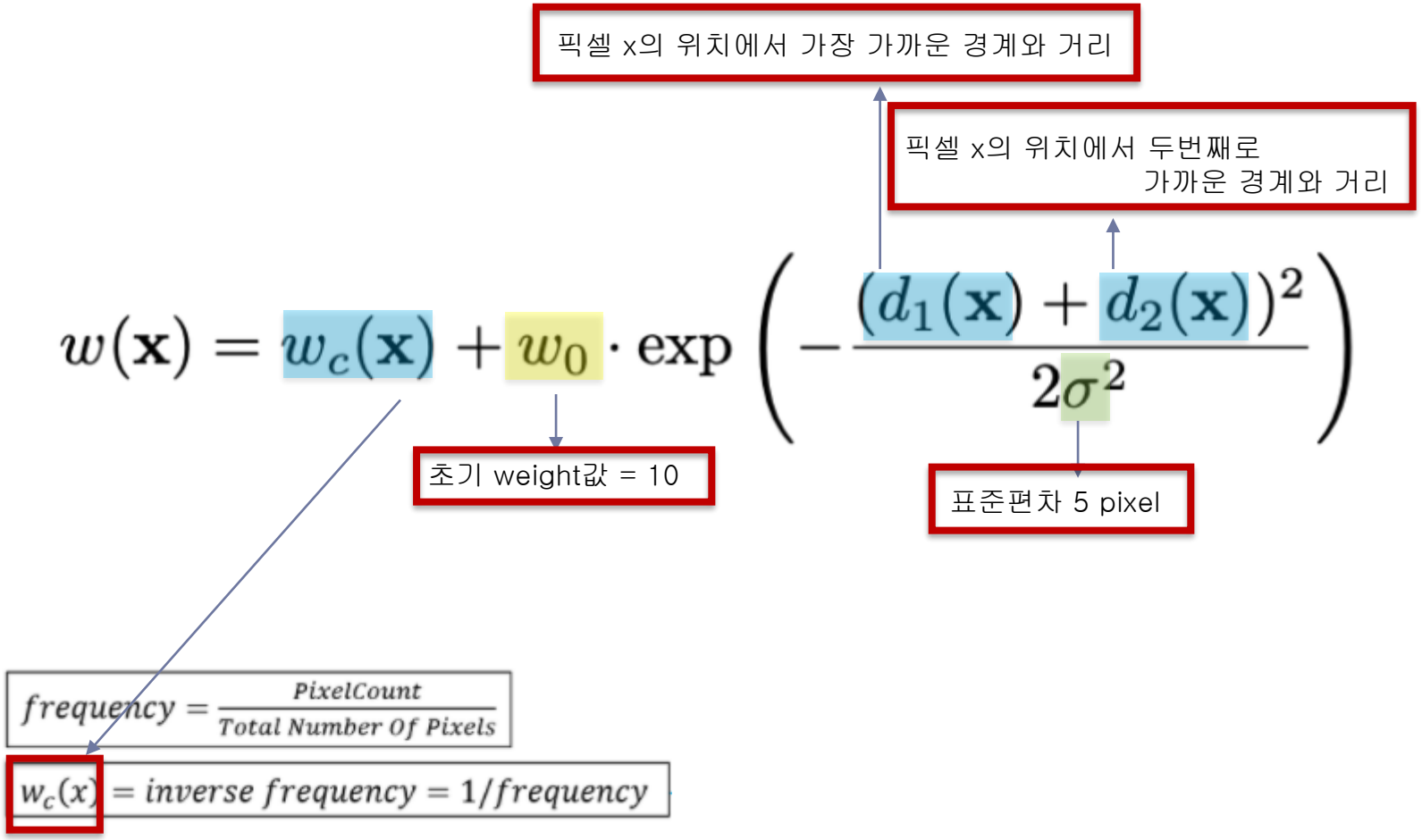


Network Architecture

- .



Training


픽셀 x의 위치에서 가장 가까운 경계와 거리
픽셀 x의 위치에서 두번째로 가까운 경계와 거리
$w(\mathbf{x}) = w_c(\mathbf{x}) + w_0 \cdot \exp\left(-\frac{(d_1(\mathbf{x}) + d_2(\mathbf{x}))^2}{2\sigma^2}\right)$
초기 weight값 = 10
표준편차 5 pixel
$frequency = \frac{PixelCount}{Total\ Number\ Of\ Pixels}$
$w_c(x) = inverse\ frequency = 1/frequency$

Training

Energy Function

= Cross entropy

Softmax

$$p_k(\mathbf{x}) = \exp(a_k(\mathbf{x})) / \left(\sum_{k'=1}^{K} \exp(a_{k'}(\mathbf{x}))\right)$$

SoftMax에서 나온 정답 레이블

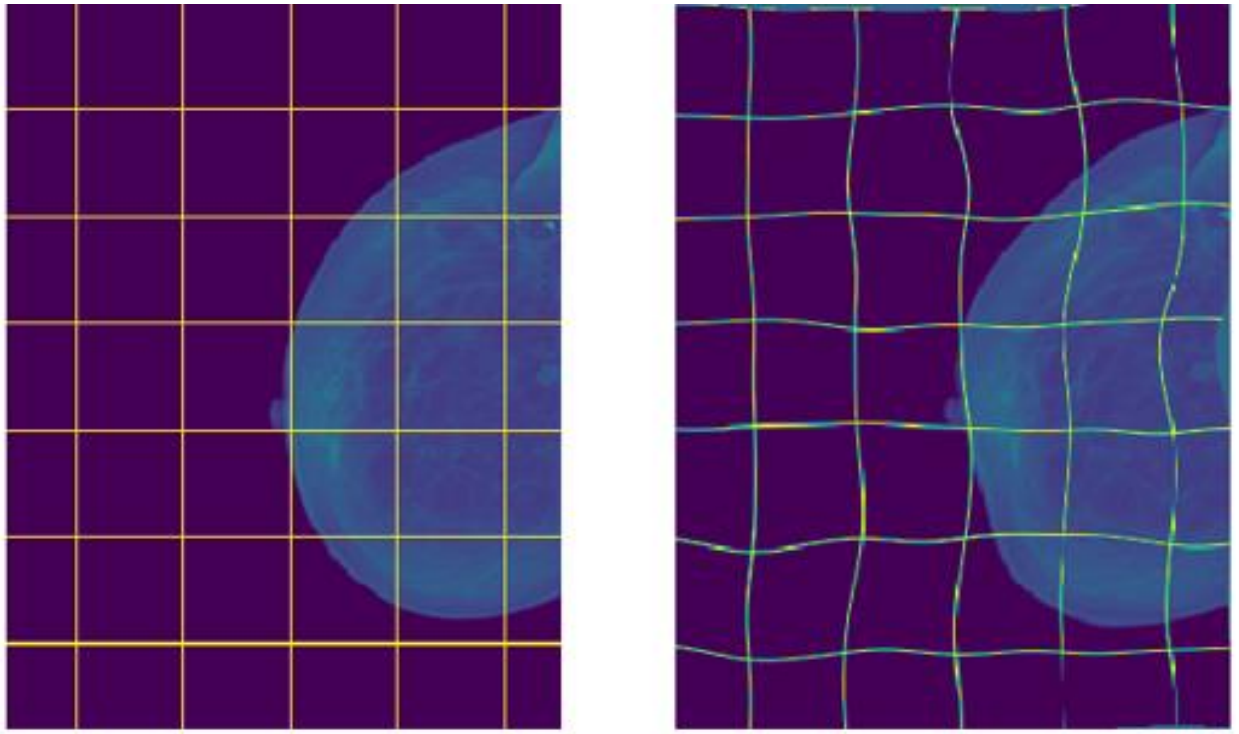
Feature map에 있는 각 픽셀

$$E = \sum_{\mathbf{x} \in \Omega} w(\mathbf{x}) \log(p_{\ell(\mathbf{x})}(\mathbf{x}))$$

Weight map : 픽셀별로 가중치를 부과한다.

Training

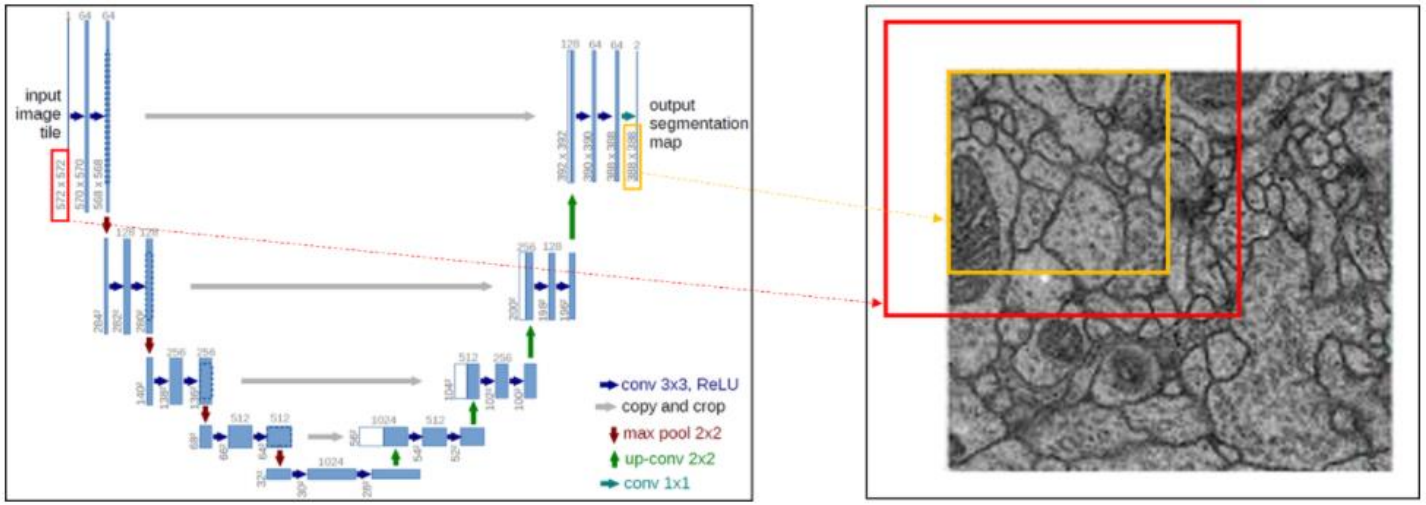
- Data Augmentation
  - Shift, Rotation and Random-elastic deformation



(a) Original (b) Deformed

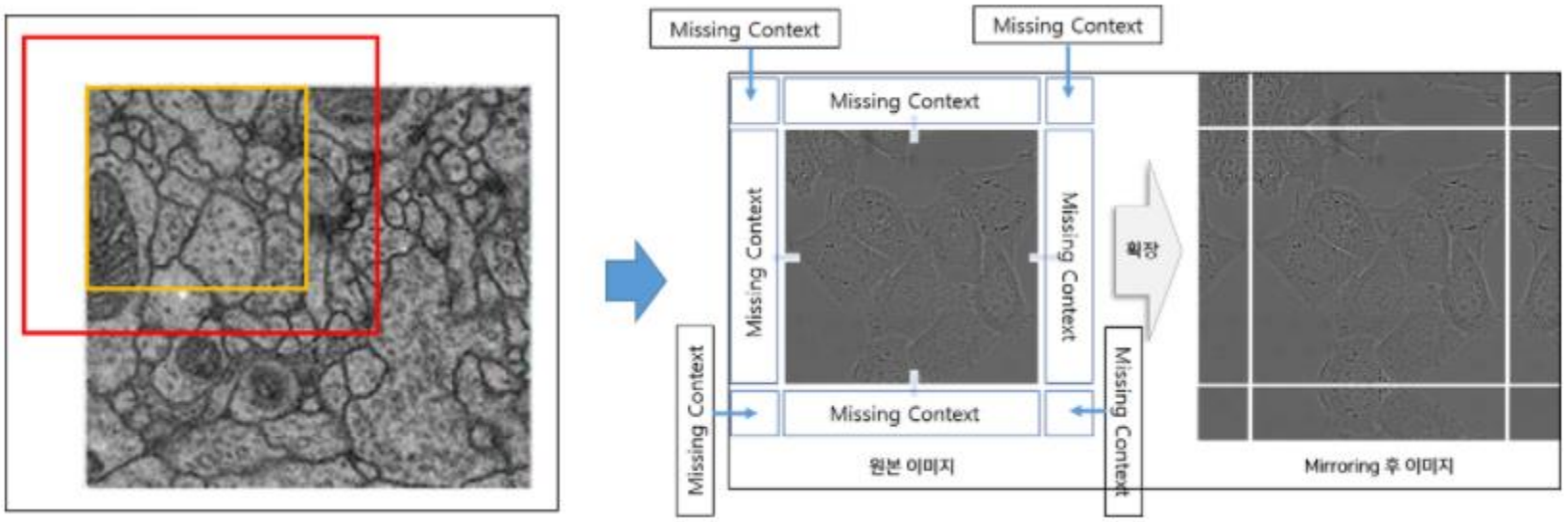
Training

- Overlap-tile strategy



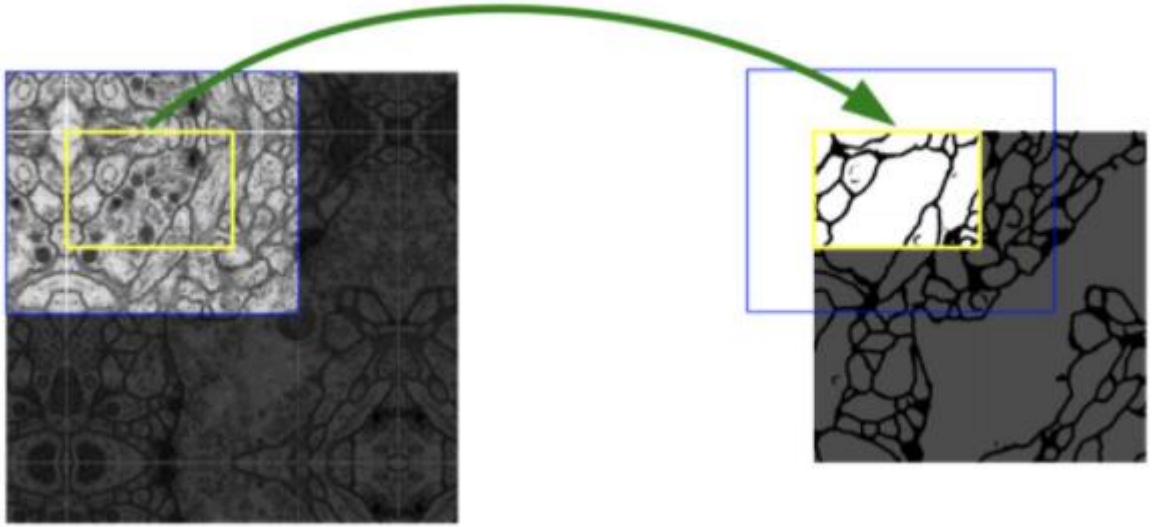
Training

- Overlap-tile strategy



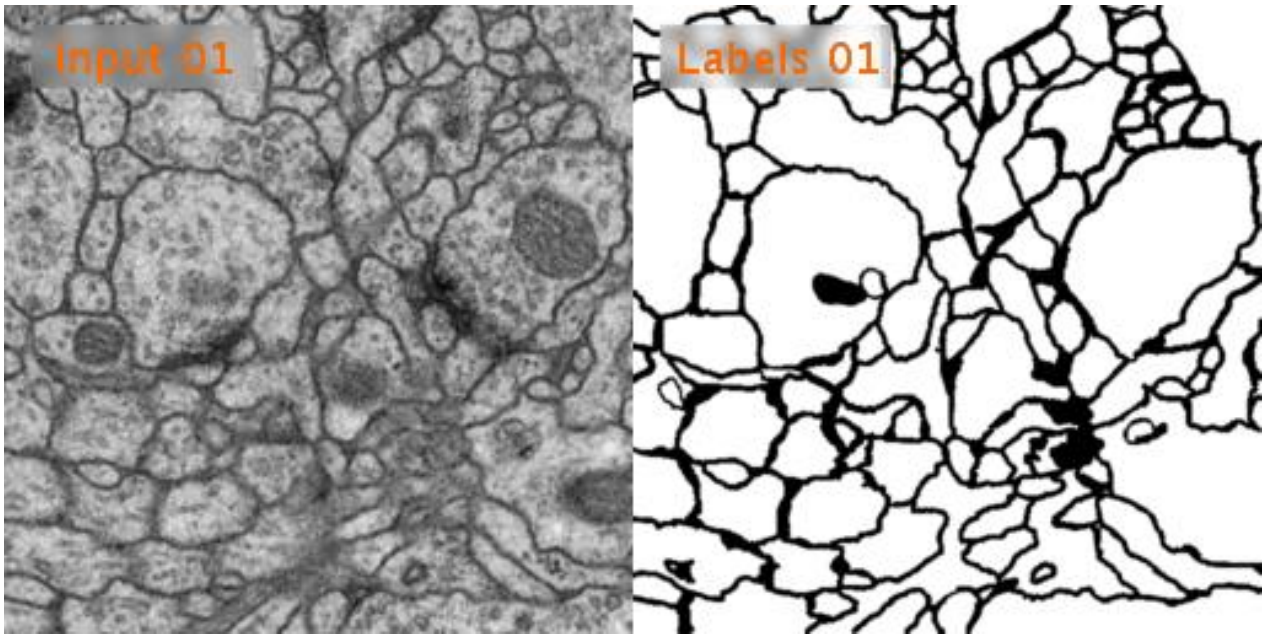
Training

- Overlap-tile strategy



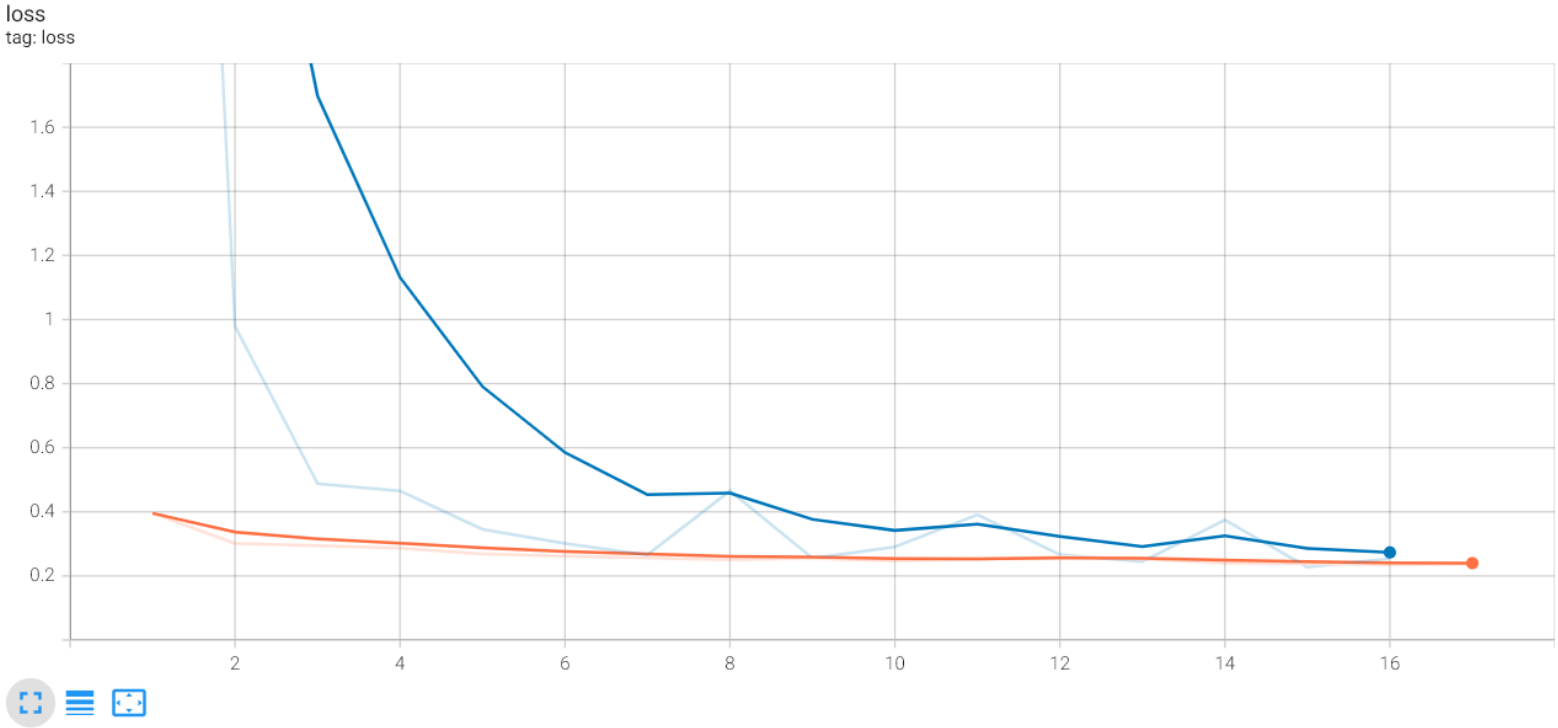
Training

- DataSet



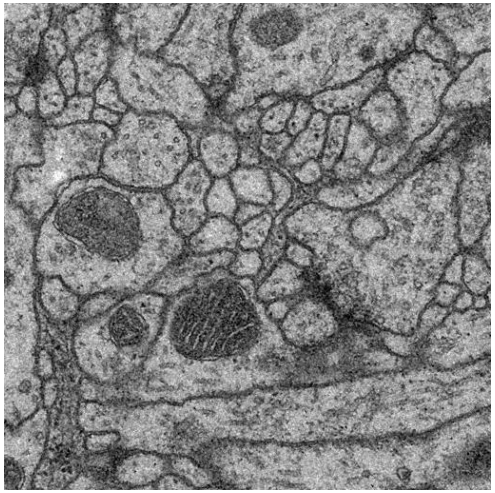
Training

- Hyper parameter
  - Lr : 10^-2, batch size : 2, epoch 100
    - Train loss : 0.15, validation loss : 0.18

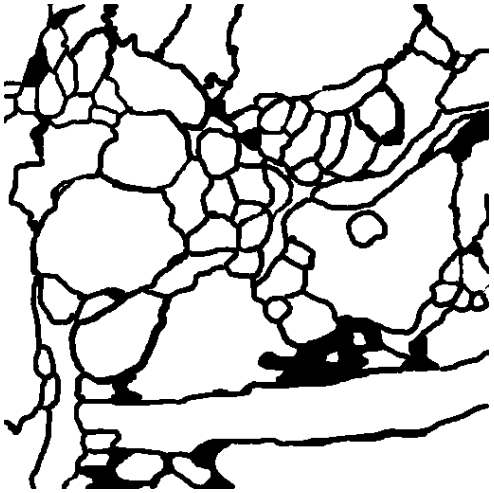


Conclusion

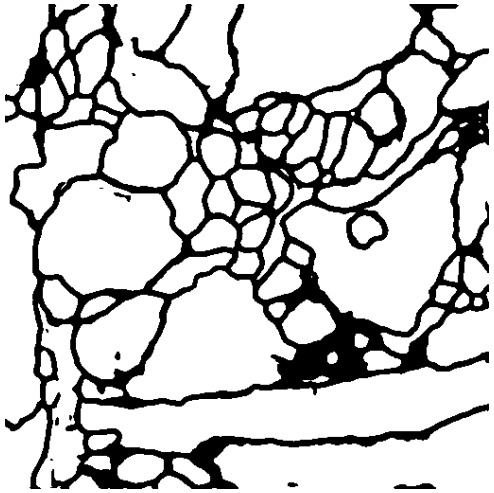
- Test



Input



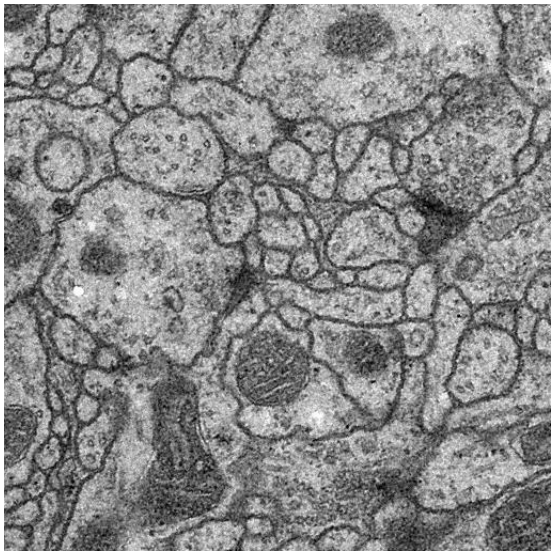
Label



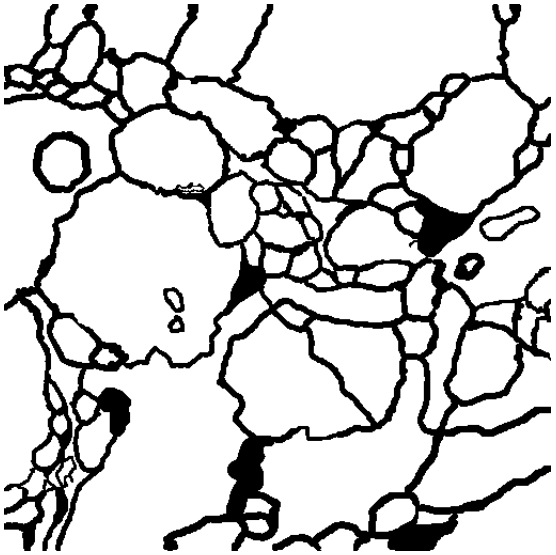
Output

Conclusion

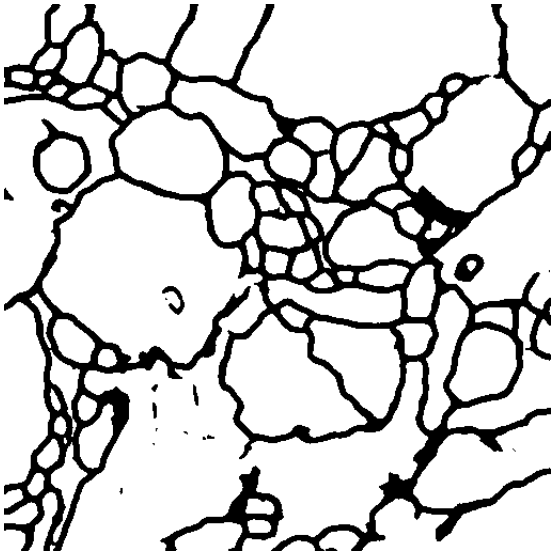
- Test



Input



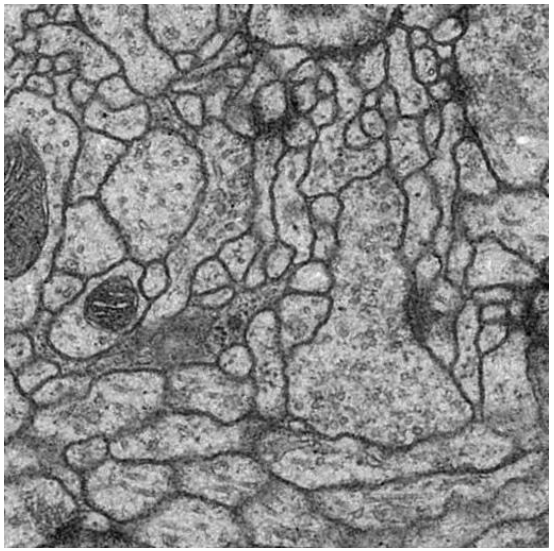
Label



Output

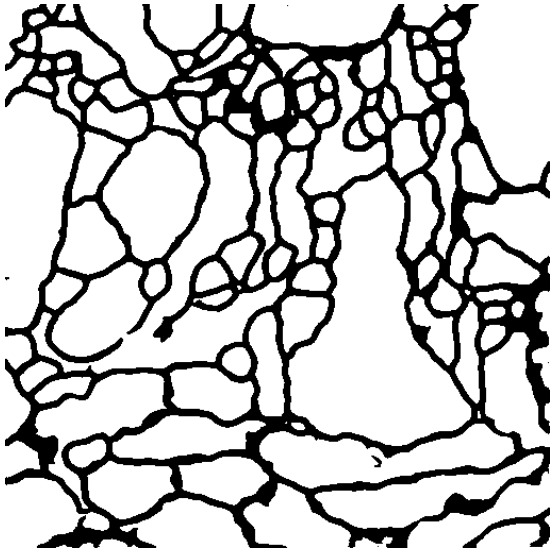
Conclusion

- Test



Input

Label



Output