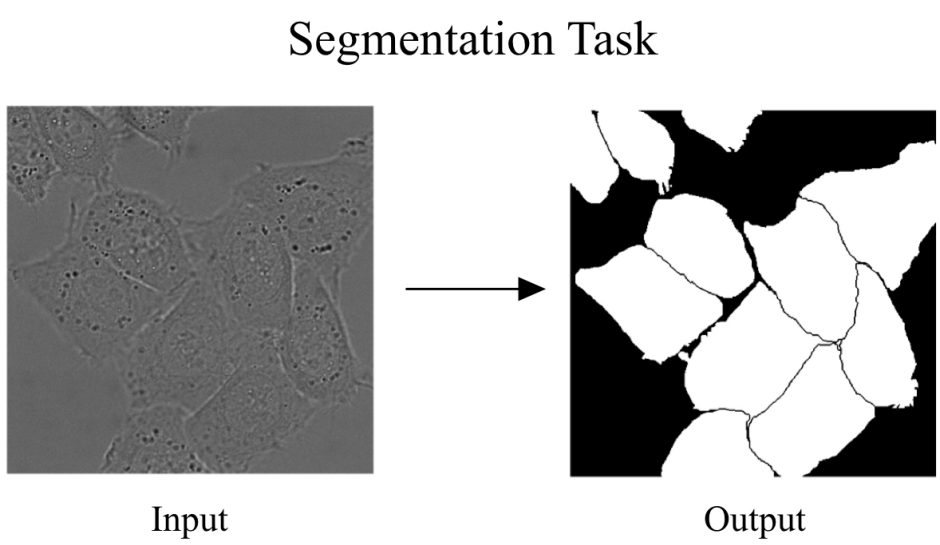
U-Net: Convolutional Networks for Biomedical Image Segmentation

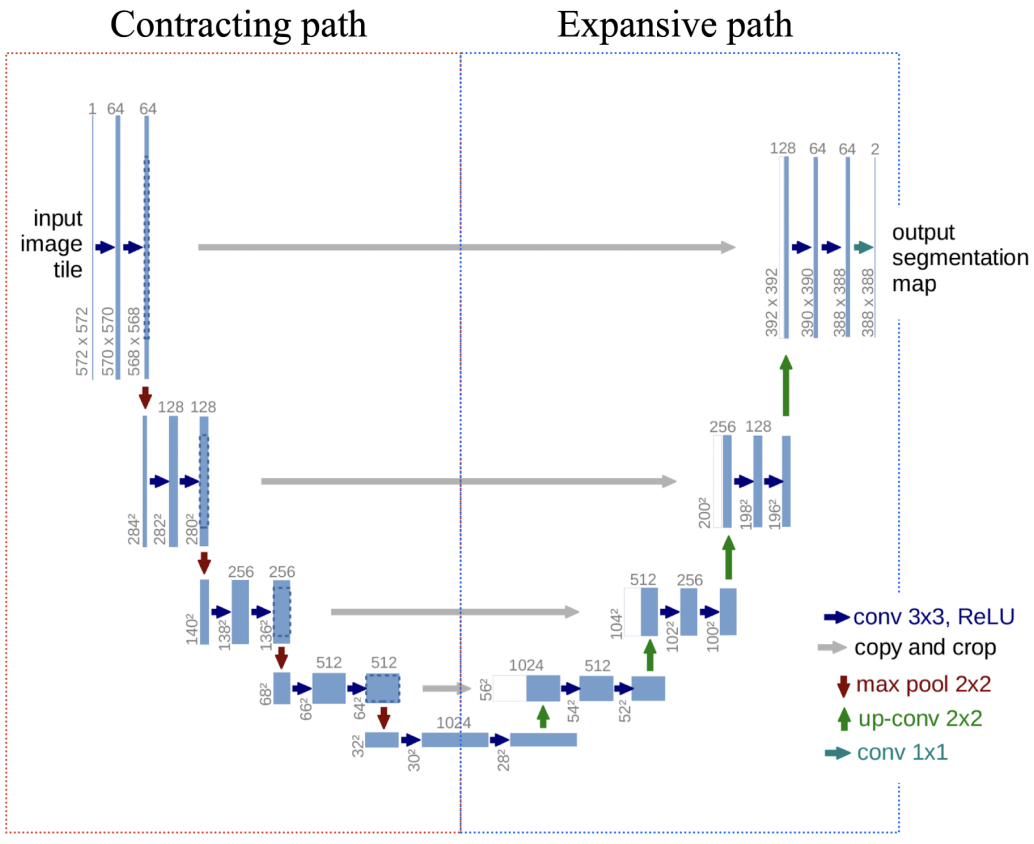
목차

1. Quick Review

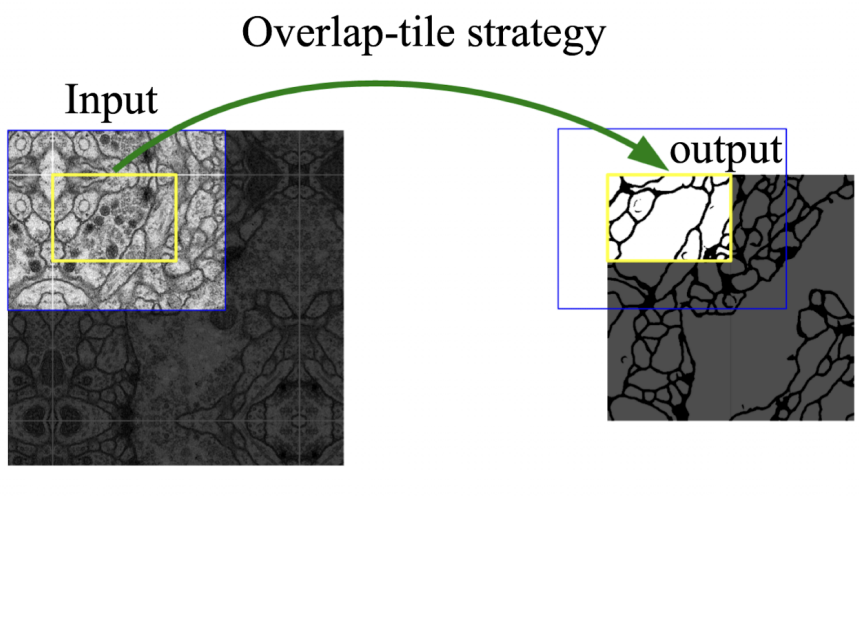
Fully-convolutional network로 이루어진 U-Net 구조를 제안하여 segmentation 분야에서 높은 성능을 달성하였다.

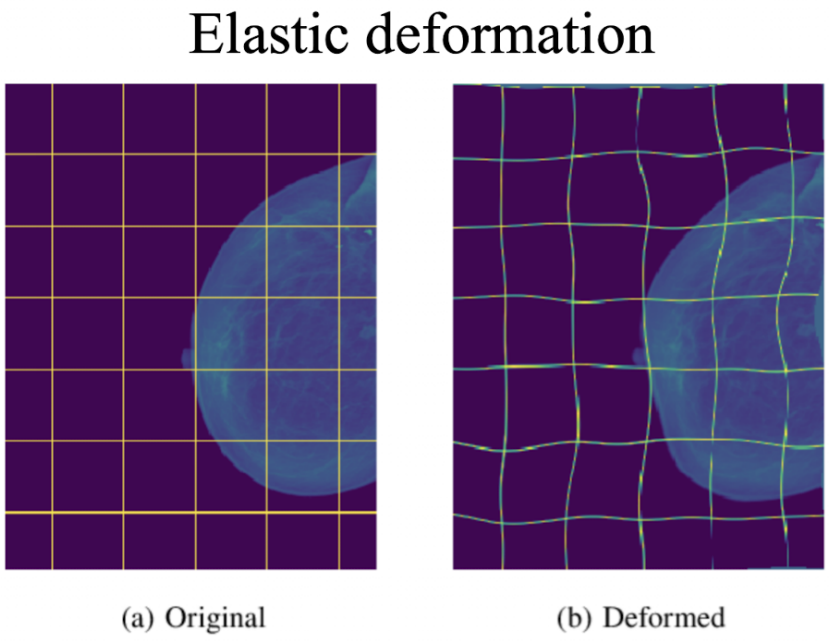


U-Net은 이름에서도 알 수 있듯이 U 형태의 네트워크 구조를 갖고 있다. Input 이미지가 들어오면 이미지의 특성을 추출하는 **contracting path**와 픽셀 단위로 예측을 하기 위해 다시 up-sampling 하는 **expansive path**가 존재한다. 일반적인 CNN 모델 구조와 달리 fully-connected layer가 존재하지 않는다.

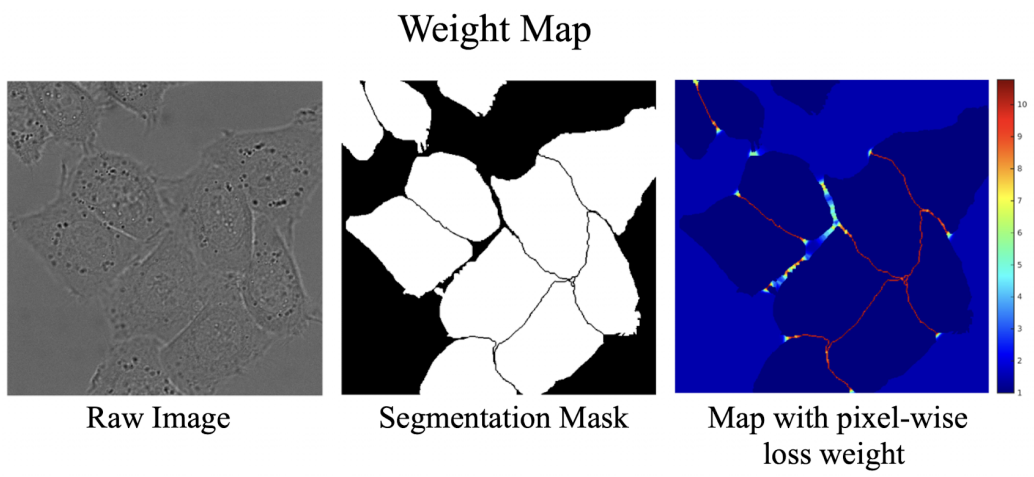


저자들은 하나의 원본 이미지에서 여러 patch를 생성하여 모델의 입력값으로 주는 **overlap-tile strategy**을 적용하였고, 데이터 증강 기법으로는 의학 분야에서 많이 쓰는 **elastic deformation**을 사용하였다.





또한 논문의 저자들은 객체 사이의 boundary를 더 잘 학습시키기 위해, boundary에 해당하는 pixel에 가중치(weight map)를 주는 방식을 사용했다.



U-Net은 biomedical 분야의 데이터를 분할하는 목적으로 제안됐지만, 현재 인용수가 20,000이 넘을 정도로 의료 영상뿐만 아니라 다양한 분야에서 활용되고 있다.

2. Introduction

논문에서는 이미지 전체를 특정 레이블로 분류하는 것이 아닌, 이미지 내의 특정 물체가 있는 영역을 검출(segmentation)하는 것을 목적으로 한다. 따라서 일반적으로 conv layer → fully connected layer 구조가 아닌, fully-convolutional network 구조를 사용하여 2012 EM segmentation challenge at ISBI의 우승을 차지한 이전의 모델[4]보다 월등한 성능을 달성했다.

특히 이전까지의 segmentation 문제에서 주로 사용된 [4]의 네트워크의 Sliding window 방식의 문제점을 언급하며 이를 개선한 U-Net의 장점을 강조했다.

- 더 빠르고, 성능이 좋다.

3. U-Net architecture

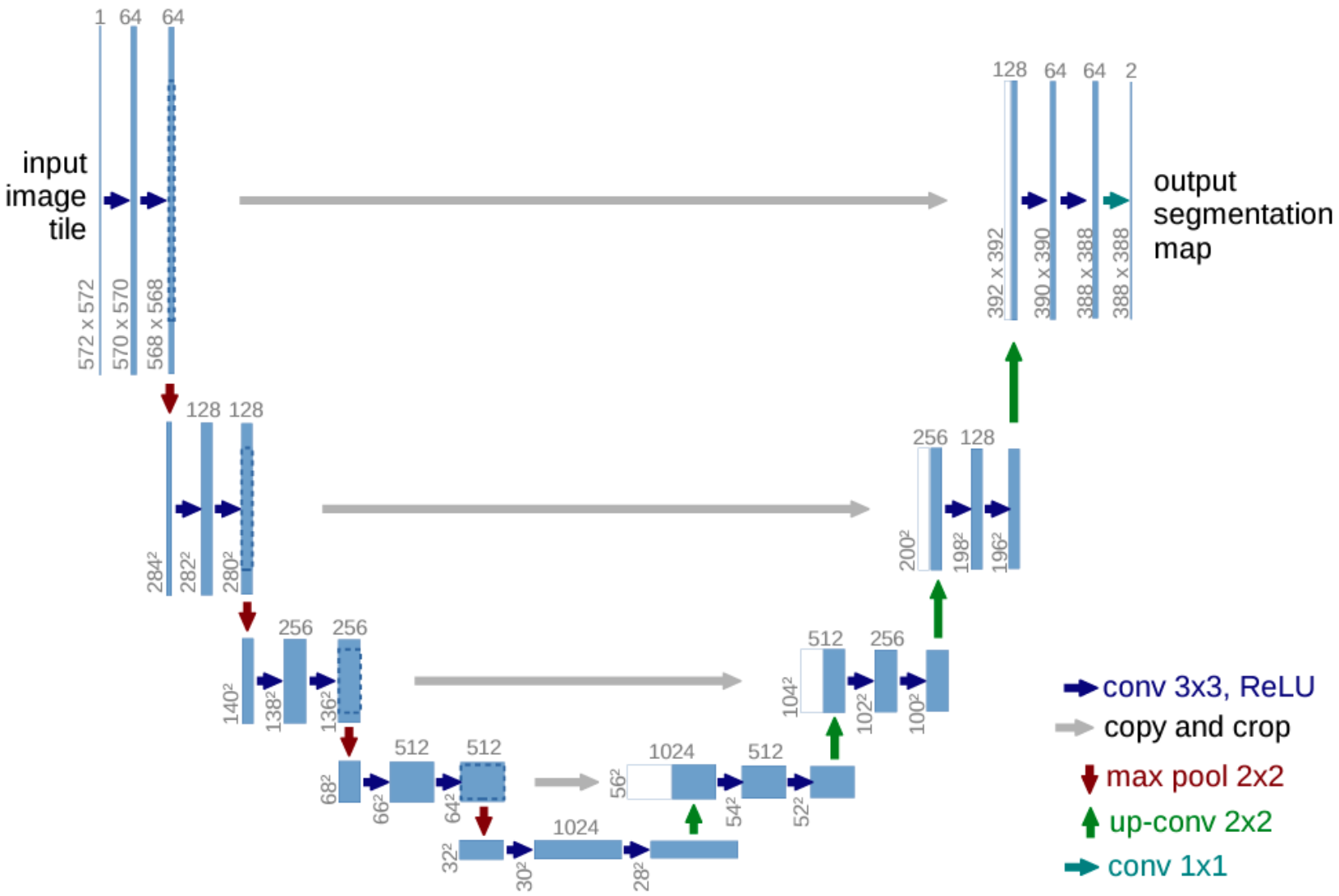
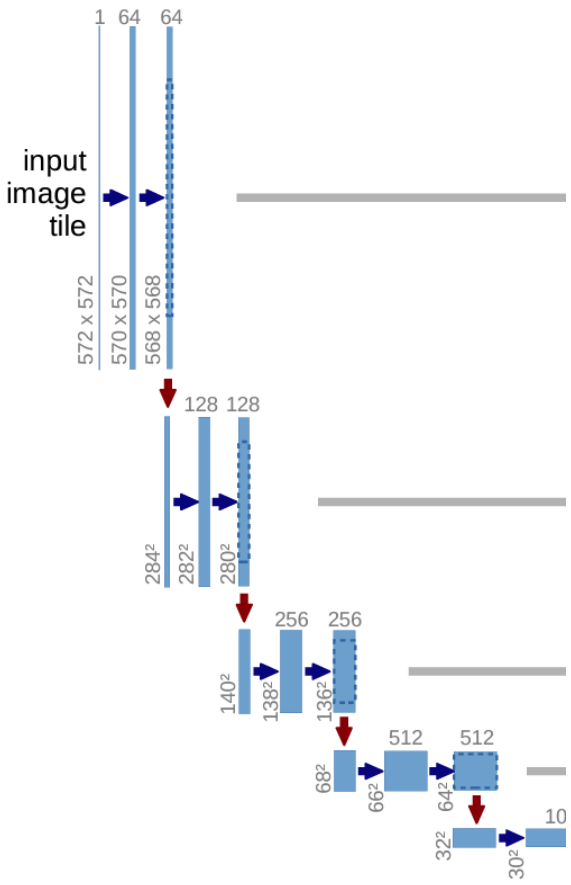


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

U-Net은 왼쪽의 **Contracting path**와 오른쪽의 **Expansive path**로 이루어져 있다.

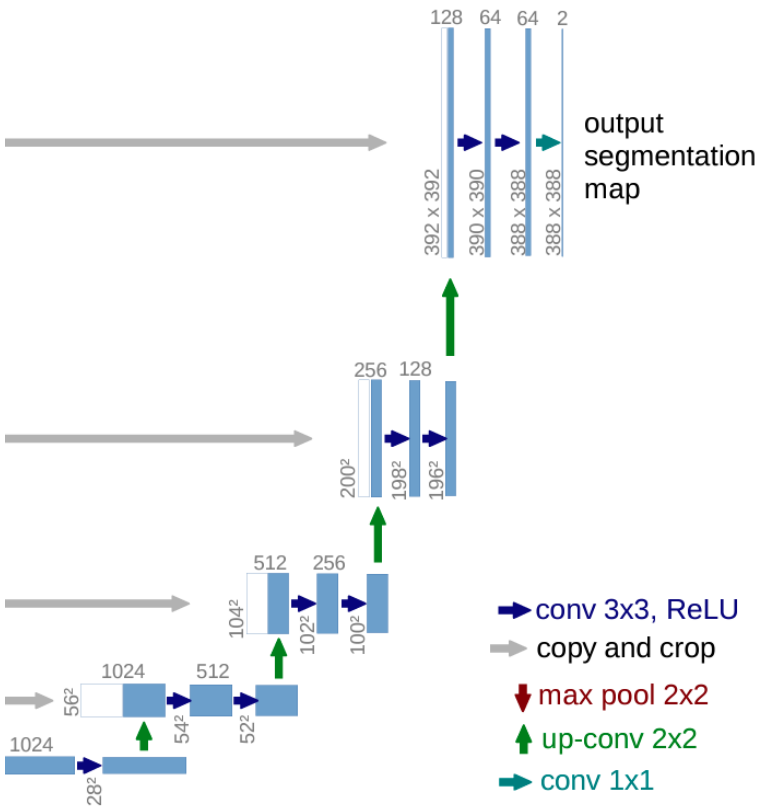
- **Contracting path**에서 이미지의 feature를 추출하고, **expansive path**를 통해 다시 높은 resolution으로 up-sampling 함으로써, 예측하고자 하는 각각의 픽셀에 대한 정보를 얻을 수 있다. 즉, 이미지의 정보를 잘 추출하고, localization을 위해 다시 원래 이미지의 크기로 복원하는 것이다. 특히 더 세밀한 localization을 위해 풍부한 정보를 사용하기 위해서, 다시 up-sampling할 때 contracting path의 feature map을 crop하여 concatenation을 수행한다.

Contracting path



- VGG-net 구조와 같다.
- 2개의 3x3 conv layer(padding=0) 사용
- 2x2 max pooling(stride=2)
- Down sampling(max pooling)시 채널이 2배가 되는 특징

Expansive path



- 2x2 up-conv layer 사용
- contracting path의 down sampling 직전의 feature map을 크기에 맞게 crop하여 concatenation 수행
- 마찬가지로 up-sampling 후 2개의 3x3 conv layer 사용
- 마지막 출력 layer에서 1x1 conv layer을 사용하여 픽셀별로 2개의 값이 추출됨.

- 이때, Input Size=(572, 572)와 Output Size=(388,388)가 다른 것을 확인할 수 있다. 이는 Overlap-tile strategy를 사용했기 때문이다. 자세한 설명은 appendix를 확인하자.

4. Training & Experiments

실험 세팅

Optimizer	Stochastic Gradient Descent(Momentum=0.99)
Loss function	Cross Entropy

학습에 적용된 추가적인 기법들

1. Cross Entropy loss with weight map

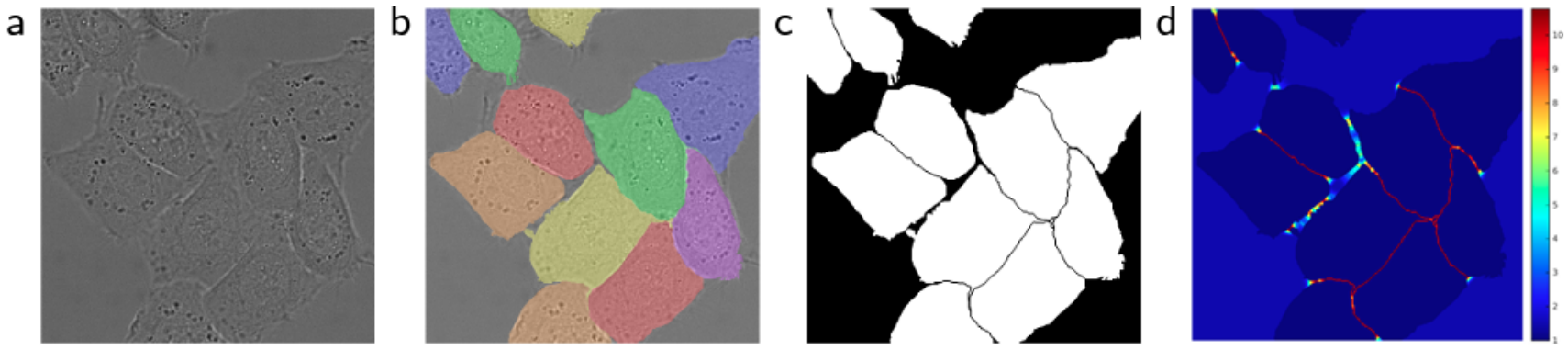


Fig. 3. HeLa cells on glass recorded with DIC (differential interference contrast) microscopy. (**a**) raw image. (**b**) overlay with ground truth segmentation. Different colors indicate different instances of the HeLa cells. (**c**) generated segmentation mask (white: foreground, black: background). (**d**) map with a pixel-wise loss weight to force the network to learn the border pixels.

a - 원본 이미지, c - ground truth label, d - weight map 시각화.

위 이미지에서 a와 c를 비교해보면, 모델이 segmentation해야하는 객체 사이의 경계가 존재한다. 하지만 객체간 경계가 전체 픽셀에서 차지하는 비중은 매우 작기 때문에, 이 픽셀들에 대해 weight를 주지 않을 경우 잘 학습되지 않아 여러개의 객체가 한개의 객체로 표시될 가능성이 높다[2]. 따라서 본 논문의 저자들은 Loss function에 이러한 경계 픽셀에 대한 weight를 주어 학습을 진행하였다.

$$E = \sum_{x} w(x)\log(p_{l(x)}(X))$$

$$\text{where } \; w(x) = w_c(x) + w_0 \cdot exp(-\frac{(d_1(x) + d_2(x))^2}{2\sigma^2})$$

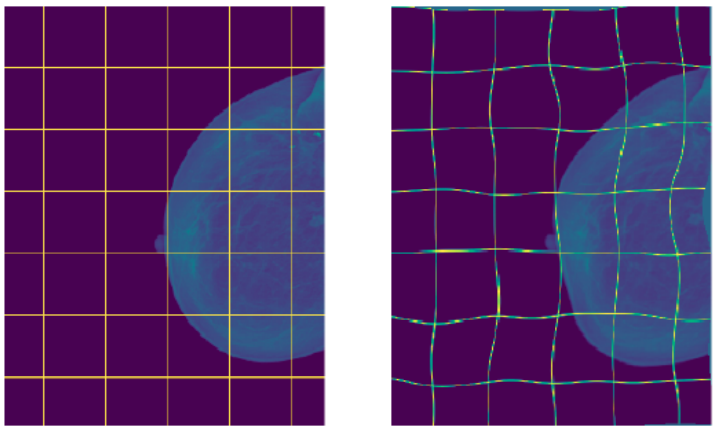
- w_c: 클래스별 픽셀의 빈도수의 균형을 맞추기 위한 weight값
- w_0, σ: weight 하이퍼 파라미터. 논문에서는 각각 10과 5로 설정.
- $d_1(x)$: 픽셀 x의 위치로부터 가장 가까운 경계와의 거리
- $d_2(x)$: 픽셀 x의 위치로부터 두 번째로 가까운 경계와의 거리

Cross entropy에 $w(x)$값이 곱해져있는데, 이는 픽셀 x와 경계의 거리가 가까우면($d_1(x) + d_2(x)$의 값이 작을수록) 큰 값을 갖게 된다. 따라서 그만큼 loss값이 커지게 되므로, 학습시 이러한 경계에 해당하는 픽셀들을 잘 학습할 수 있게 된다.

2. Data augmentation

적은 데이터셋을 학습하기 위해서는 데이터 증강이 필수적이다. 저자들은 elastic deformations를 사용하였다고 한다.

- **Elastic deformations**



(a) Original (b) Deformed

Fig. 3: Effects of performing elastic deformation on a mammogram.

의학 분야에서 주로 사용하는 데이터 증강 기법이다[5]. 살아있는 세포의 특성을 고려한 증강 기법이다. 본 논문에서도 해당 기법을 따로 자세하게 설명하고 있지 않고 있다. 코드 구현 방법이 궁금하다면 캐글 노트북 [6]을 참고해보자.

5. Result

- 2015 EM segmentation challenge에서 Warping Error와 Pixel Error 성능면에서 가장 우수했다.

Table 1. Ranking on the EM segmentation challenge [14] (march 6th, 2015), sorted by warping error.

Rank	Group name	Warping Error	Rand Error	Pixel Error
	** human values **	0.000005	0.0021	0.0010
1.	u-net	**0.000353**	0.0382	0.0611
2.	DIVE-SCI	0.000355	0.0305	0.0584
3.	IDSIA [1]	0.000420	0.0504	0.0613
4.	DIVE	0.000430	0.0545	**0.0582**
⋮				
10.	IDSIA-SCI	0.000653	**0.0189**	0.1027

- ISBI cell tracking challenge 2015에서 가장 우수한 IoU 성능을 보여 우승을 차지했다.

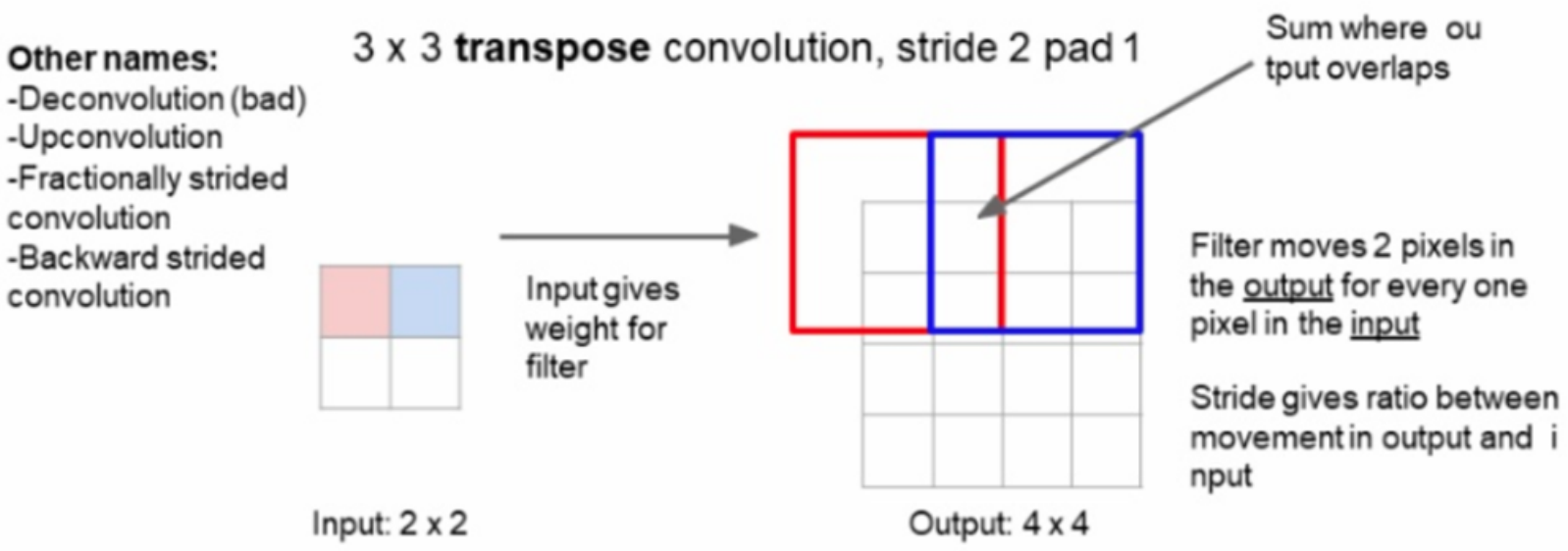
Table 2. Segmentation results (IOU) on the ISBI cell tracking challenge 2015.

Name	PhC-U373	DIC-HeLa
IMCB-SG (2014)	0.2669	0.2935
KTH-SE (2014)	0.7953	0.4607
HOUS-US (2014)	0.5323	-
second-best 2015	0.83	0.46
u-net (2015)	**0.9203**	**0.7756**

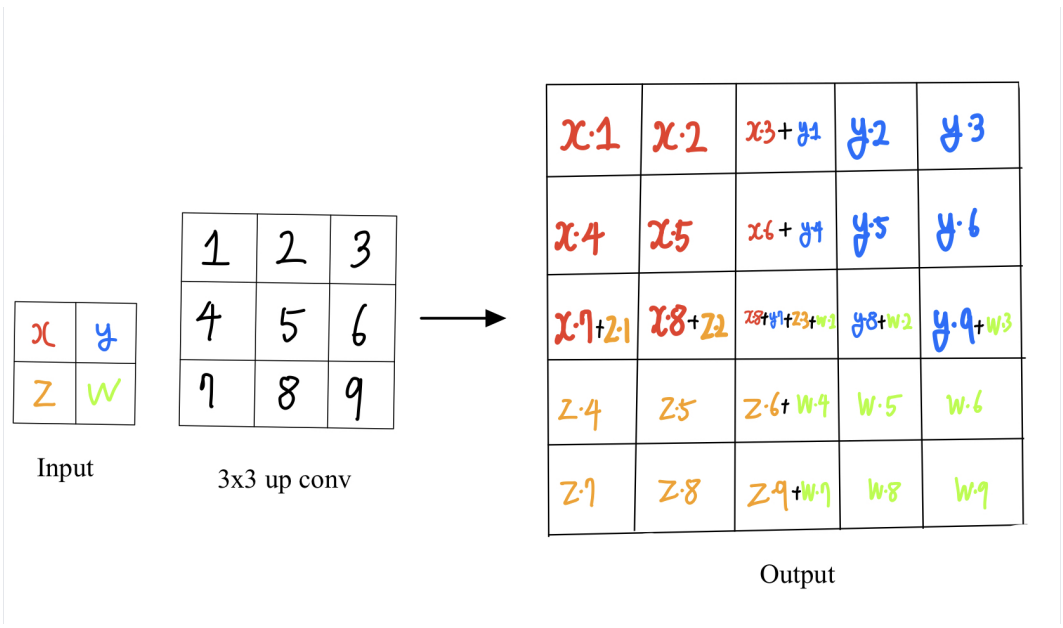
6. Appendix

- **Transpose Convolution(Up-conv)**

  Learnable Upsampling 방법이다.



예시) stride=2



위와 같이 학습 가능한 convolutional 필터를 사용하여 더 큰 resolution의 output을 생성한다.

PyTorch: `torch.nn.ConvTranspose2d`

- **Ciresam et al의 Sliding Window 방식**

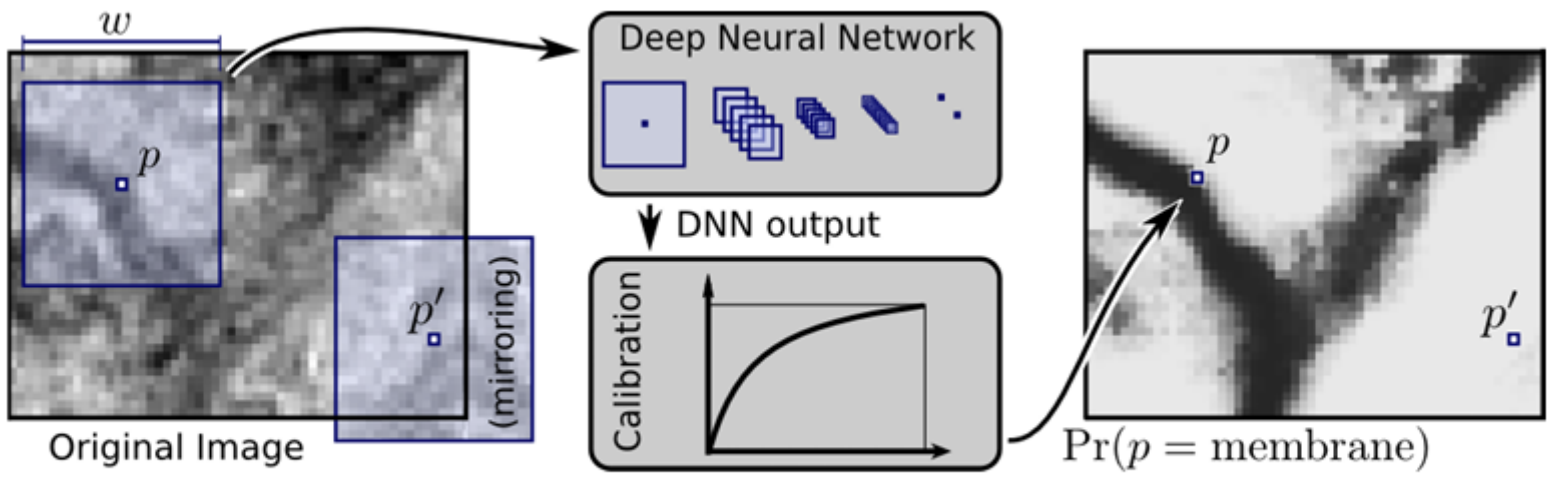


Figure 2: Overview of our approach (see text).

모든 픽셀에 대해 패치를 생성하여 학습하는 방식이다. Sliding window 방식에서는 해당 픽셀이 object인지 background인지 분류하는 DNN classifier를 학습한다. 해당 논문에서는 학습에 512x512 크기의 이미지 30장을 사용하였는데, 한 이미지당 object에 해당하는 픽셀이 평균 50,000개라고 한다. 따라서 한 이미지당 약 50,000개의 ground truth label=1인 인풋 이미지가 만들어진다. 레이블 분포를 맞추기 위해 background 인풋 이미지(label=0)는 label≠0 에 해당하는 픽셀에서 비슷한 개수로 랜덤 샘플링하여 추출했고 한다[4].

- 모든 패치에 대해서 네트워크가 학습해야하기 때문에 Train은 물론 test할 때도 상당히 느릴 것이라고 예상할 수 있다.
- 또한 이 논문에서 가장자리쪽에 있는 픽셀에 대해 mirroring 방법론을 적용하였다. U-Net 논문의 저자도 여기서 아이디어를 가져오지 않았을까 싶다.(주관)

- **Overlap-tile Strategy**

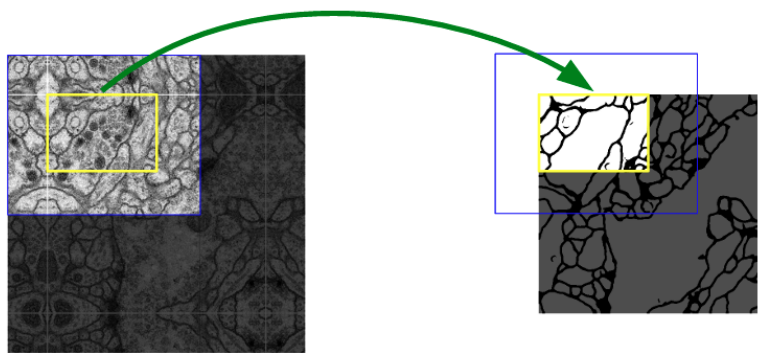
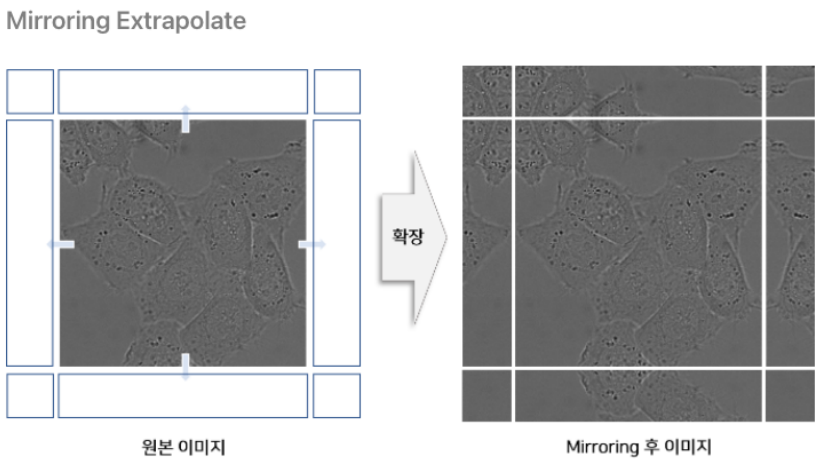


Fig. 2. Overlap-tile strategy for seamless segmentation of arbitrary large images (here segmentation of neuronal structures in EM stacks). Prediction of the segmentation in the yellow area, requires image data within the blue area as input. Missing input data is extrapolated by mirroring

논문 제목에서도 알수 있듯, 본 논문에서는 의료 영상 데이터인 전자 현미경 데이터를 다루고 있다. 이러한 현미경 데이터 특성상 원본 이미지 사이즈가 상당히 크기 때문에, patch 단위로 잘라서 Input으로 넣었다고 한다.

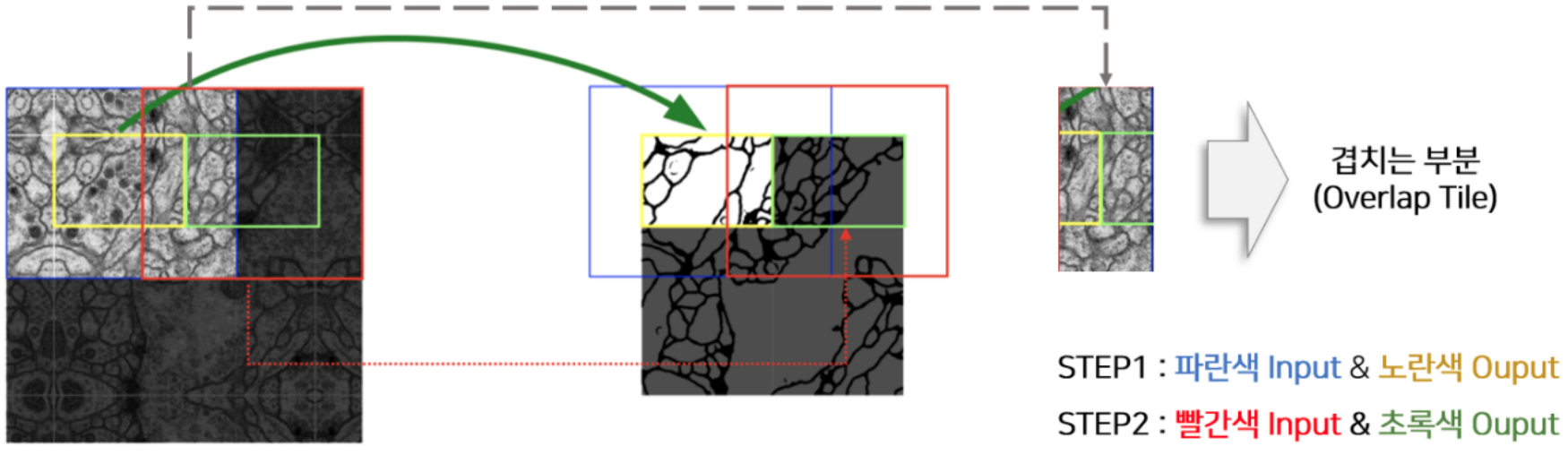
U-Net의 구조를 보면 Conv layer에서 padding=0으로 설정하기 때문에, Input shape이 (572, 572)인 반면, Output shape은 (388, 388)로 축소된 것을 알 수 있다. Fig. 2에서 **파란색 영역**이 Input이 되는 것이고, **노란색 영역**이 output으로 출력되는 부분이 되는 것이다.

따라서 더 노란색 영역을 segmentation하기 위해 파란색 영역으로 만들어 네트워크의 입력으로 넣어야 하는데, 원본 이미지 바깥 부분에 대해서는 Mirroring 방법으로 pixel 값을 채워 patch를 생성하였다.



이미지 출처: [2]

여기서 Overlap이라고 부르는 이유는, 아래와 같이 다음 입력 데이터를 생성할 때 겹치는 부분이 발생하기 때문이다.



이미지 출처: [2]

- Rand error, Warping error, Pixel error
  - Segmentation metric이다.

Rand error: defined as $1 - F_{\text{rand}}$, where F_{rand} represents the F_1 score of the *Rand index* [29], which measures the accuracy with which pixels are associated to their respective neurons.

Warping error: a segmentation metric designed to account for topological disagreements [19]; it accounts for the number of neuron splits and mergers required to obtain the candidate segmentation from ground truth.

Pixel error: defined as $1 - F_{\text{pixel}}$, where F_{pixel} represents the F_1 score of pixel similarity.

[4]에 기재된 설명이다. 자세한 설명은 [7]을 참고하자.

References

- [1] CS231n Winter 2016: Lecture 13: Segmentation, soft attention, spatial transformers
- [2] https://joungheekim.github.io/2020/09/28/paper-review/
- [3] https://modulabs-biomedical.github.io/U_Net
- [4] https://proceedings.neurips.cc/paper/2012/file/459a4ddcb586f24efd9395aa7662bc7c-Paper.pdf
- [5] https://stat-cbc.tistory.com/28
- [6] https://www.kaggle.com/ori226/data-augmentation-with-elastic-deformations
- [7] https://ashm8206.github.io/2018/04/08/Segmentation-Metrics.html
- [8] https://realblack0.github.io/2020/05/11/transpose-convolution.html