

Masking (Resampling) Techniques

Wednesday, 18 September 2019

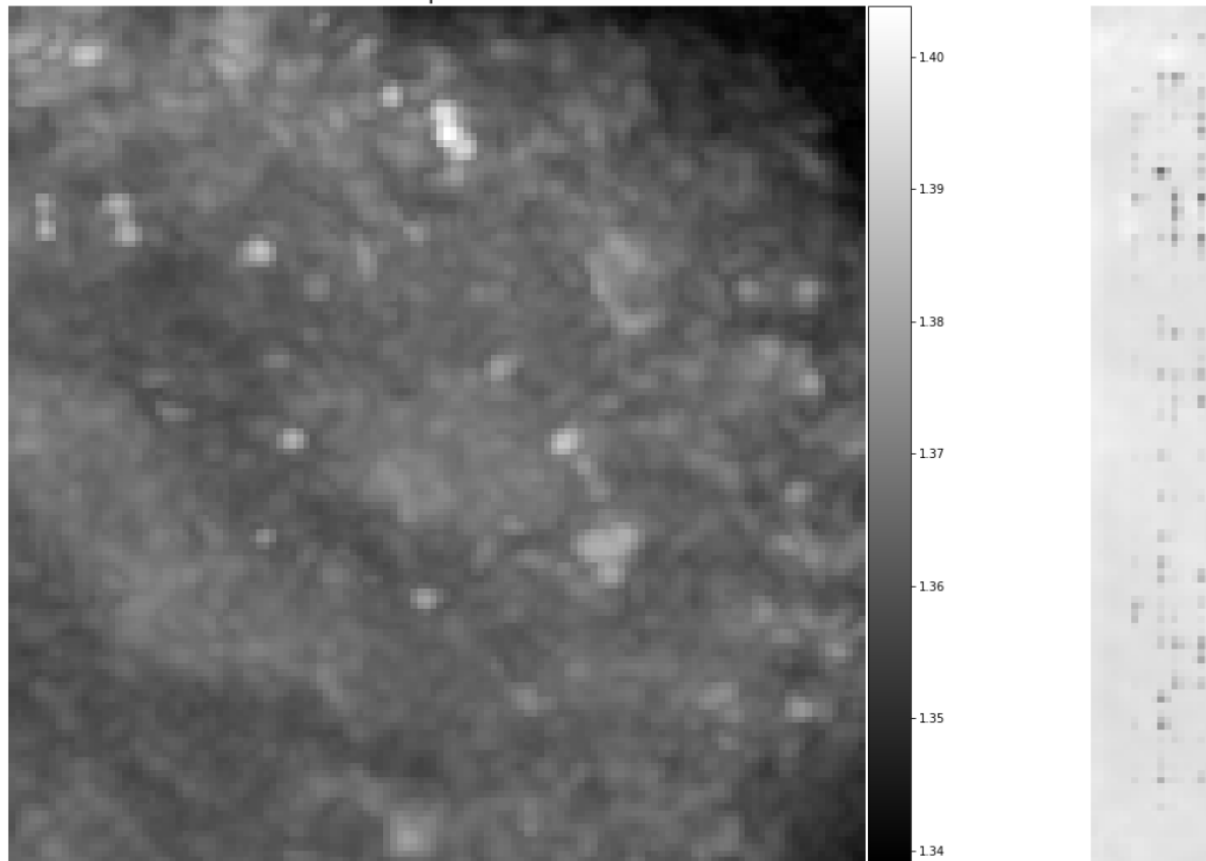
1:02 PM

- 사용중인 모델: Noise2Void (<https://arxiv.org/abs/1811.10980>)
 - 목적: 정답이 없는 상황에서의 노이즈 제거
 - 원리: 인접한 픽셀들의 시그널에는 상관관계가 있지만, 노이즈는 독립적
 - 주의사항: 모델이 identity를 학습하지 못하도록 input과 target 사이에 차이를 주어야 한다.
 - 해결책: input image의 일부 픽셀을 다른 값으로 바꾼다.
 - 형주, 동민의 8월 27일 실험(1번서버 /data3/noise2meeum)에서는 일부를 변경

- 문제 상황:

outs/jaeun/n2v_tomocube_data/20190723.155953.040.Default-037_MIP_128_320.hdf

Input

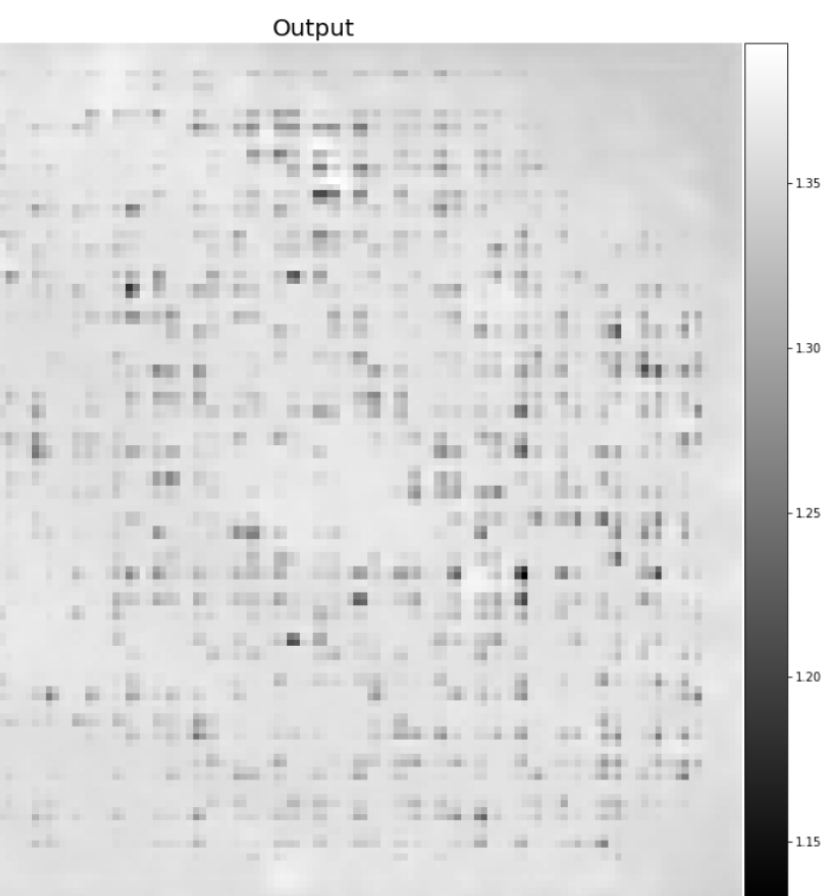


○

- 토모큐브 3D RI 데이터의 MIP를 입력값으로 삼았을 때 효과가 없다.

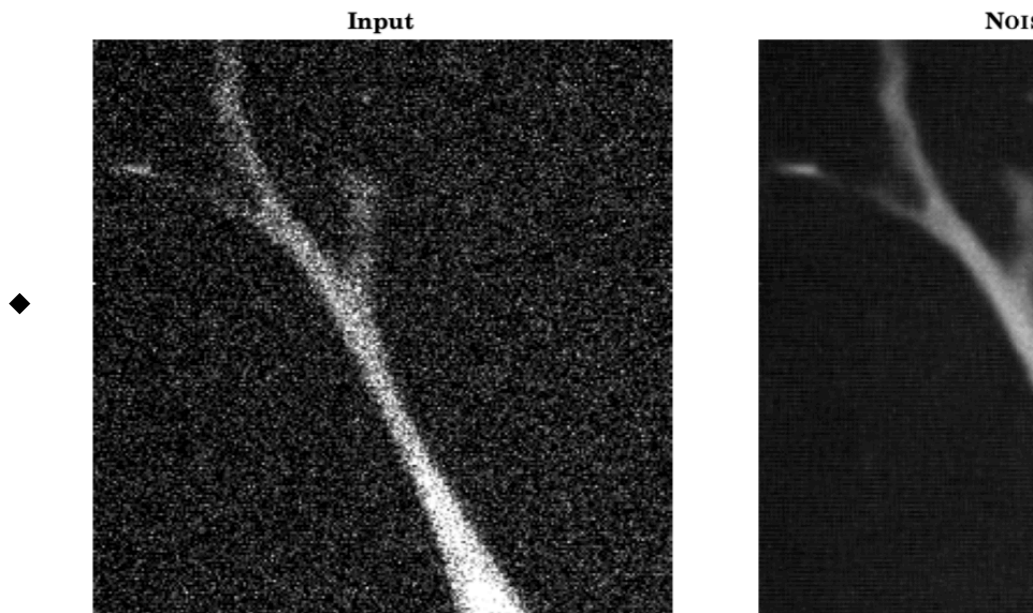
이다.
자이를 뒤야 한다.

Uniform Pixel Selection 방식으로 input image의

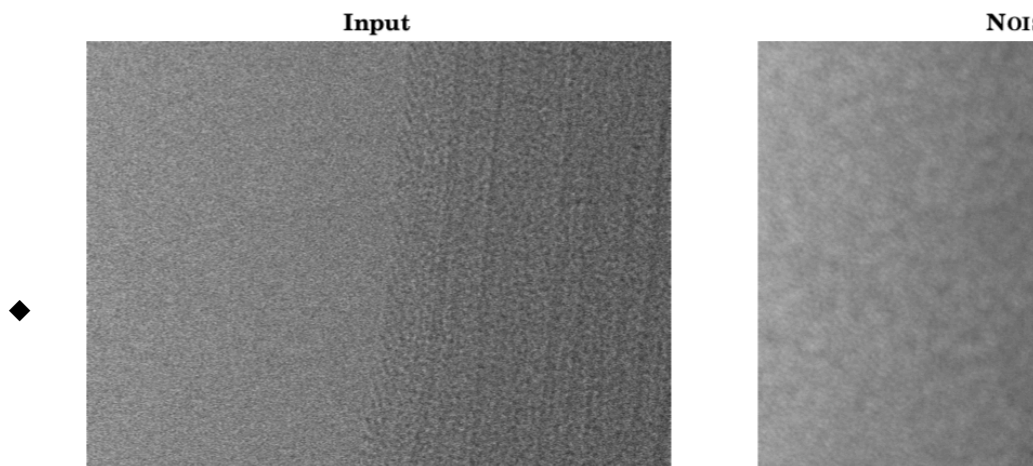


○ 관찰되는 문제점

- RI 값의 범위가 넓어지고 평균이 낮아짐. Input range: 1.34~1.40(평균 1.27, 범위 0.25)
- 색이 반전됨. 즉, 배경의 RI 값이 높게 측정됨.
 - N2V supplementary material에서
- 격자모양 artifact가 생김.
 - 예상 원인: input image에 있던 격자모양 artifact가 N2V를 거쳐
 - N2V supplementary material 중 CTC-MSC dataset(cell track)의 artifact가 생긴것을 볼 수 있다.
 - http://openaccess.thecvf.com/content_CVPR_2019/supplemental/supplemental.pdf



- Cryo-TEM dataset에서 원래는 input image의 왼쪽 절반이 균일 구조가 생긴 것을 볼 수 있다.



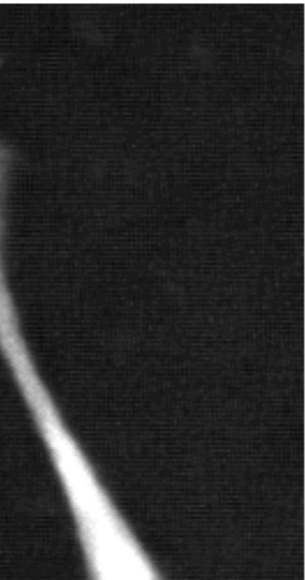
평균 1.37, 범위 0.06), Output range: 1.14~1.39(평균

치며 강화되었다.

ing challenge에서 제공하는 데이터)에서도 격자무

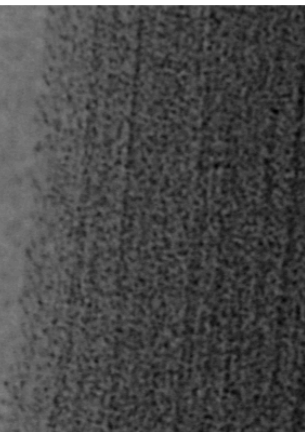
[/Krull Noise2Void - Learning CVPR 2019](#)

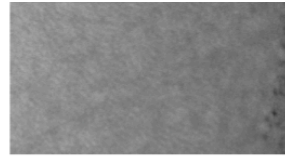
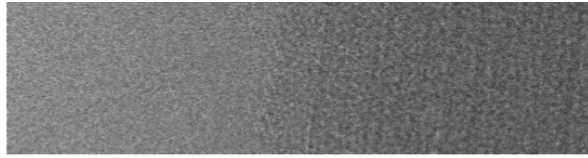
SE2VOID



관일한 노이즈였으나 N2V를 거치고 나면 거시적인

SE2VOID





- 가설

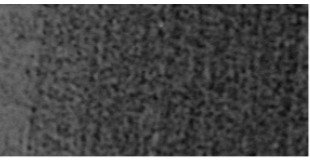
- N2V는 특정 픽셀이 인접 픽셀들과 시그널을 공유한다고 가정하였다. 그 가정이 깨지게 된다. 이때는 인접 픽셀이 시그널 값에 대한 힌트를 주지
 - 확인 방법: input image를 resizing하여 UPS를 적용해본다. 만약 in 사하였는데도 결과에서 똑같은 격자 무늬가 나타난다면 이 가설은
- Shot noise외의 다른 노이즈가 존재할 수 있다. 예를 들면, Nonnegativity 노이즈가 생길 수 있다. N2V가 이 패턴을 강화하는 것으로 보인다.
 - 확인 방법: 빈 화면에 gaussian noise를 더해 '배경+노이즈' 이미지 무늬가 생기지 않는다면 이 격자무늬는 ri image를 찍는 과정에서
 - Supplementary material 중 CTC-MSC에서 왜 격자무늬 패턴이 생겼는지
- 진행 상황: N2V 논문의 supplementary materials를 참고하여 Gaussian, Gaussian을 추가로 구현 http://openaccess.thecvf.com/content_CVPR_2019/supplemental/supplemental.pdf

3.1 Masking Methods

We tried the following variants of our masking scheme: **Uniform Pixel Selection (UPS)** replaces the value of the selected pixel i with a randomly selected pixel value from a square window around i . This includes the pixel itself. In Supplementary Table 1 we write the size of the window in parenthesis.

Gaussian (G) masking changes the value of the selected pixel i by adding random Gaussian noise. In Supplementary Table 1 we provide the standard deviation of the noise in parenthesis.

- The **Gaussian Fitting (GF)** method looks at a local square neighborhood of the pixel i and fits a 1D Gaussian distribution to the pixel values by calculating mean and standard deviation. The pixel i is included in this calculation. The pixel value is



런데 만약 시그널의 크기가 픽셀 하나 정도라면 이 못하므로 모델이 제대로 학습될 수 없다.

put image 픽셀 하나의 값을 4개, 9개의 픽셀에 복
울지 않다.

y preprocessing으로 인해 격자무늬 패턴을 가진

를 모사한다. 이것에 UPS를 적용하였을 때는 격자
생긴 것이다.

겼는지 생각해볼것

sian Pixel Selection, Gaussian Fitting 세 가지 방법

[Krull Noise2Void - Learning CVPR 2019](#)

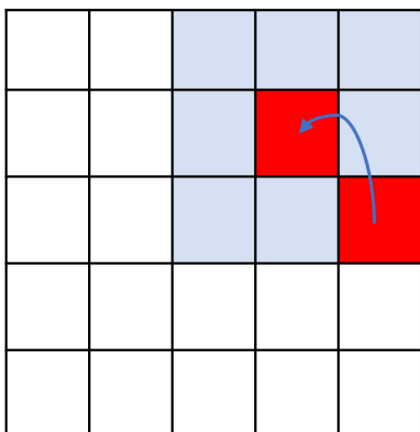
masked by drawing from the fitted Gaussian distribution. In Supplementary Table 1 we write the size of the neighborhood in parenthesis.

The **Gaussian Pixel Selection (GPS)** replaces the value of the selected pixel i with a randomly selected pixel that is found by

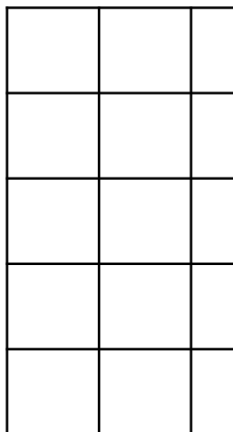
drawing from a 2D isotropic Gaussian distribution centered at pixel i . The pixel i itself is excluded from this draw. That is

- if the pixel i is selected by the Gaussian draw, we redraw until another pixel has been selected. We use a Gaussian distribution with a standard deviation of $\sigma = 4$.

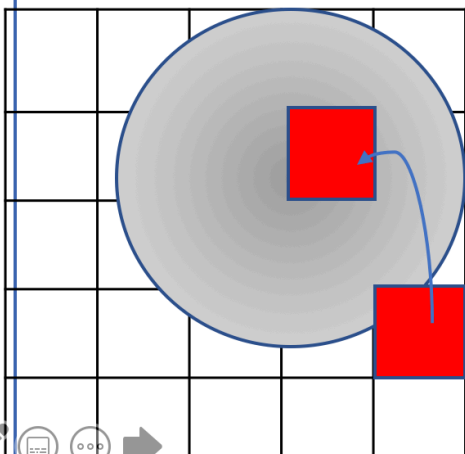
Uniform Pixel Selection (UPS)



New args:
size_mask

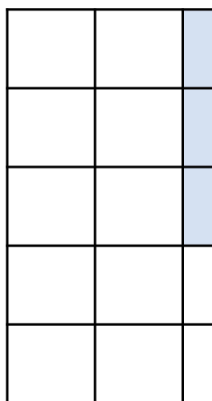


Gaussian Pixel Selection (GPS)



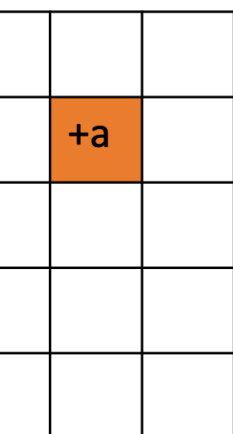
2D isotropic Gaussian:
Closer pixels have
higher chances of
being chosen

New args:
mask_std



t
s,
il
n

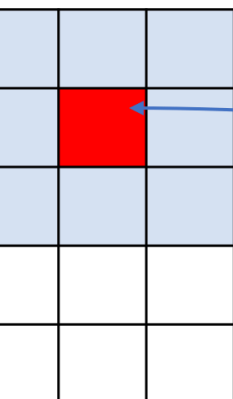
Gaussian (G)



Add a noise to the original pixel. The noise is drawn from normal distribution of mean: 0 and std: *mask_std*

New args:
mask_std

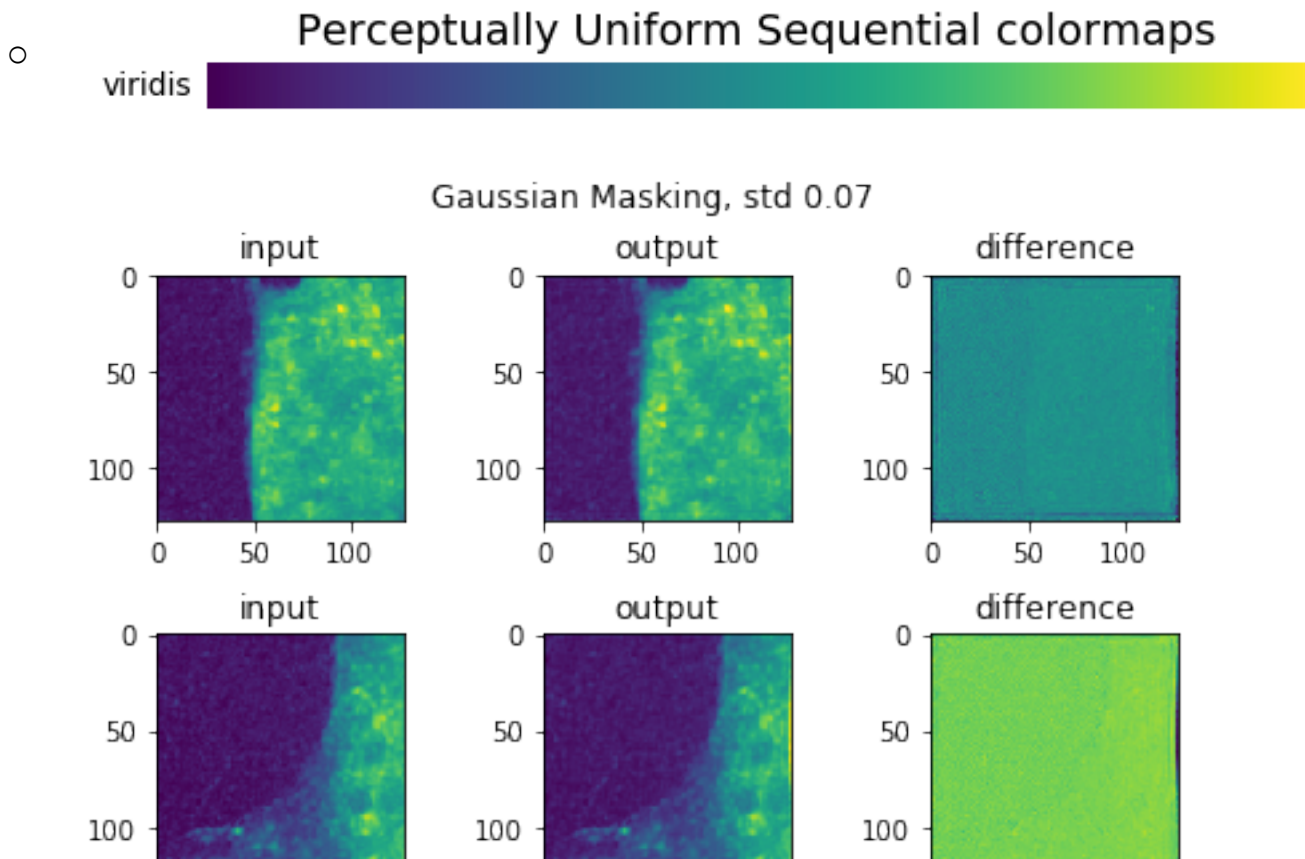
Gaussian Fitting (GF)



Mean
& std

New args:
size_mask

- masking을 사용하기 위해서는 ini 파일에 hyperparameter 세 가지를 입력해오
 - masking = "G" or "GPS" or "GF" or "UPS" (Default: UPS)
 - mask_std: default 값 없음
 - Mask_size: default 값 없음
- size_mask와 size_std를 바꿔가며 실험
- 결과 저장 위치: [0번서버](#) /data1/jaeun/n2v_masking/outs/jaeun/
 - Gaussian, 다양한 std 사용:
 - ./g_masking/: std 0.07사용. 이것이 input image의 std이기 때문
 - ./g_masking_014/: std 0.14 사용 (input std의 2배)
 - ./g_masking_003/: std 0.03 사용 (input std의 절반)
 - Gaussian Pixel Selection:
 - ./gps_masking/: mask_std 5 사용. 논문에서 mask_std가 5였을때 가
한 현미경의 배율이 우리 데이터와 다를 수 있기 때문에 다양한 m
- 결과
 - 각 디렉토리의 0.hdf와 1.hdf가 결과 파일. 각 파일은 'input'과 'output'을
 - Output viewer: [0번서버](#) /data1/jaeun/n2v_masking/Jaeun's output view

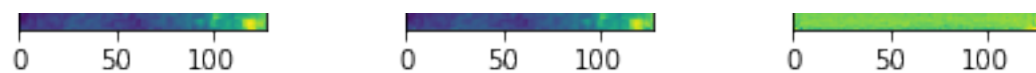


야 한다.

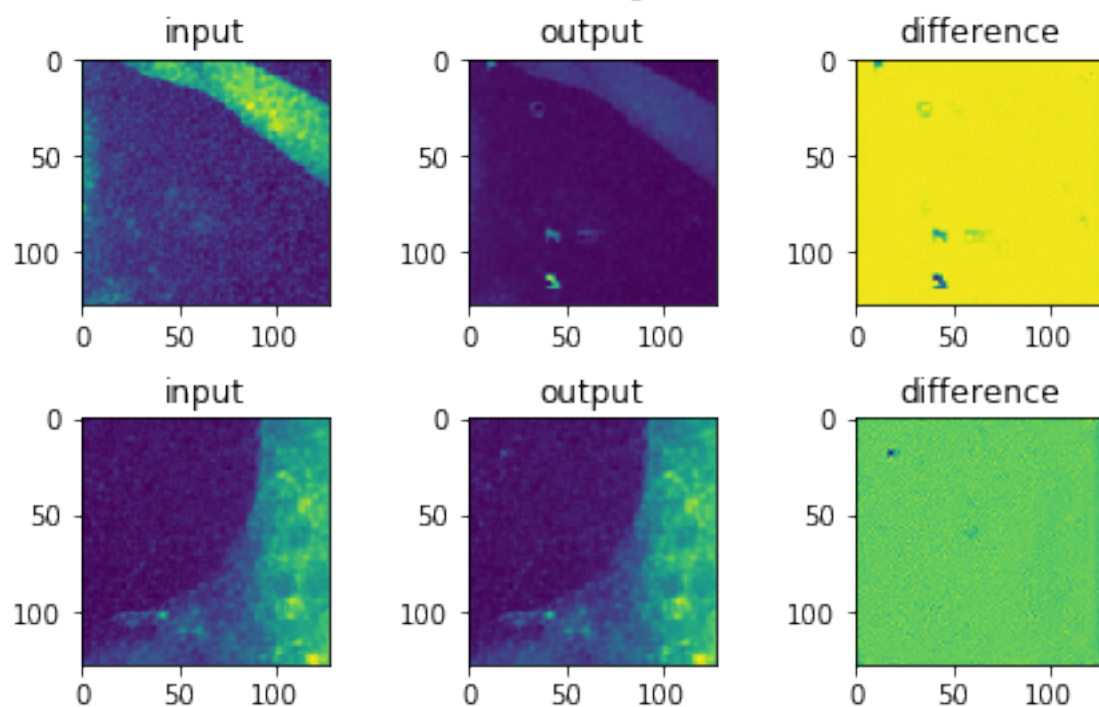
가장 좋은 결과가 나왔다고 했기 때문. 하지만 사용
mask_std 값을 사용해봐야 한다.

을 가지고 있음

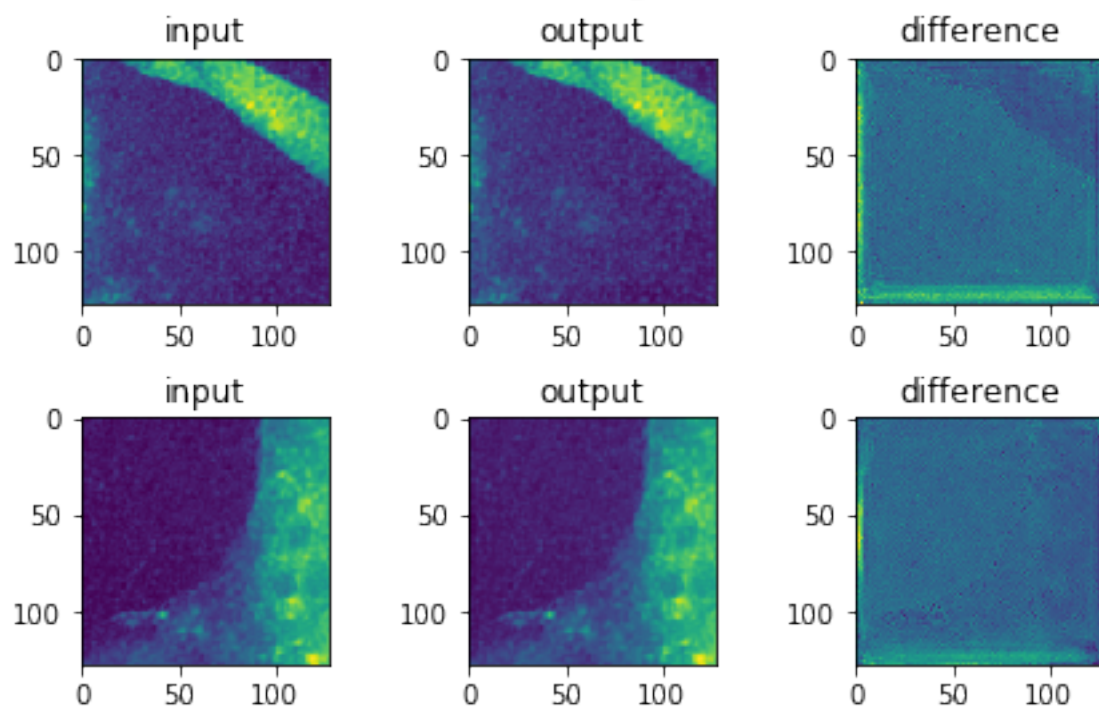
ver.ipynb



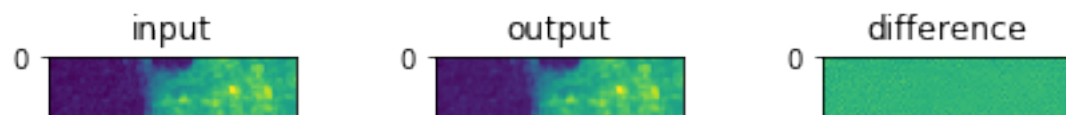
Gaussian Masking, std 0.14

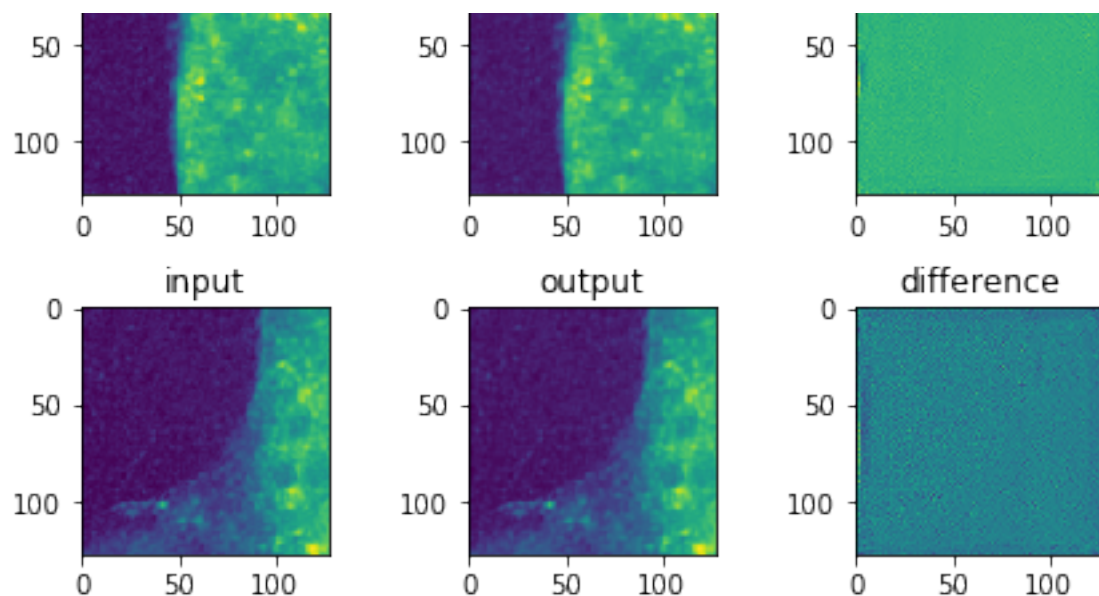


Gaussian Masking, std 0.03

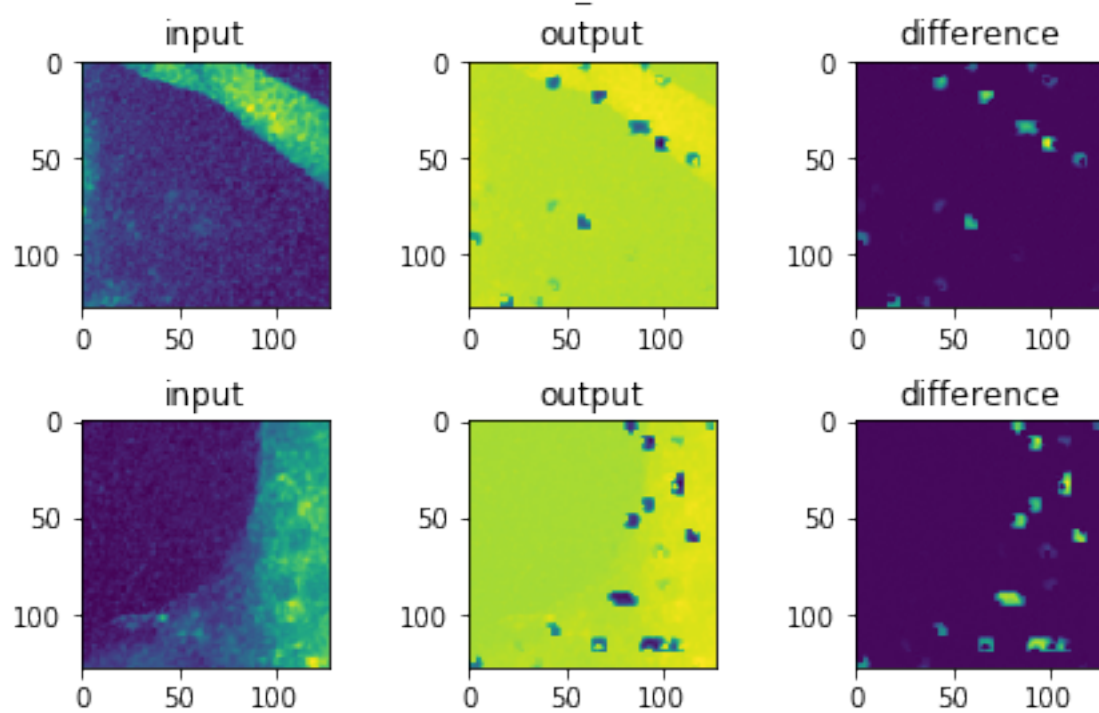


Gaussian with single SCNAS output channel

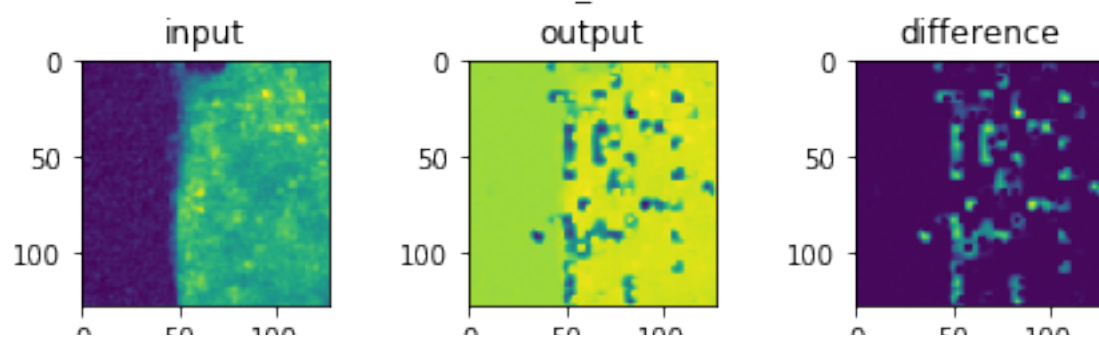


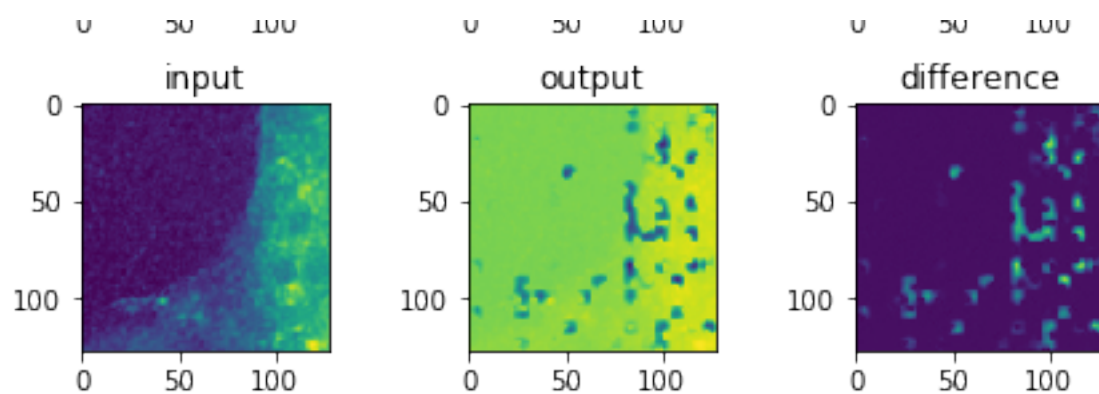


GPS: size_mask=3

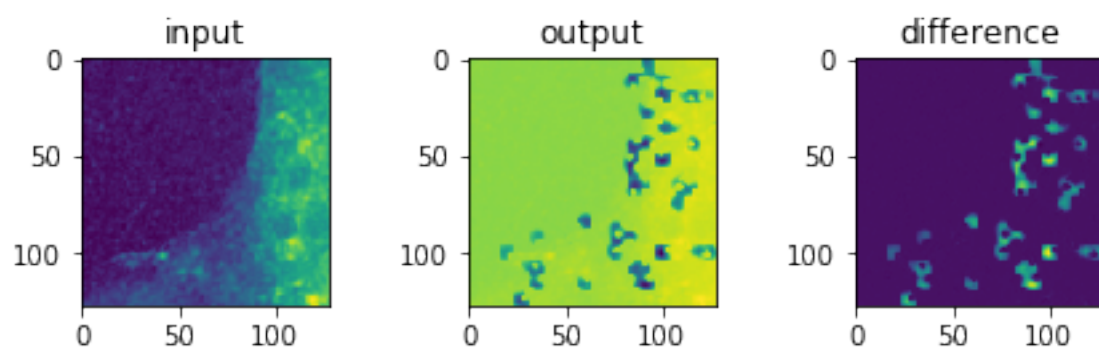
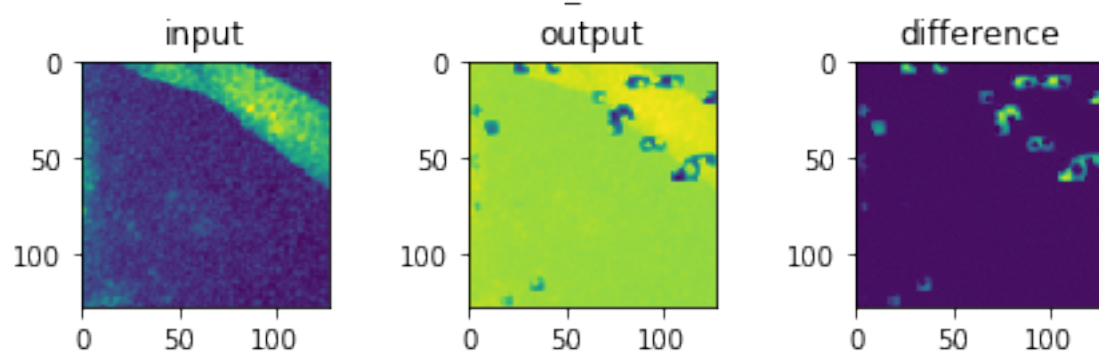


GPS: size_mask=5

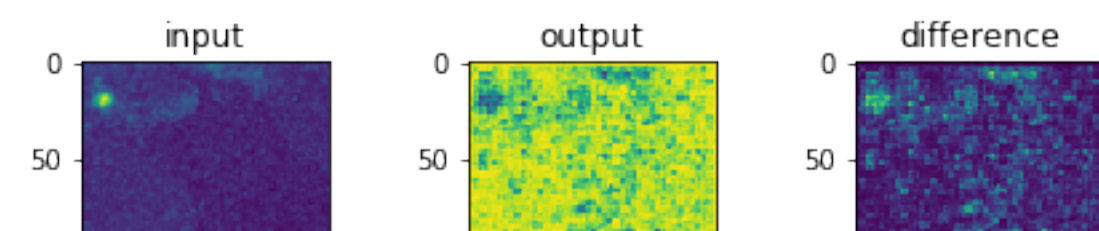
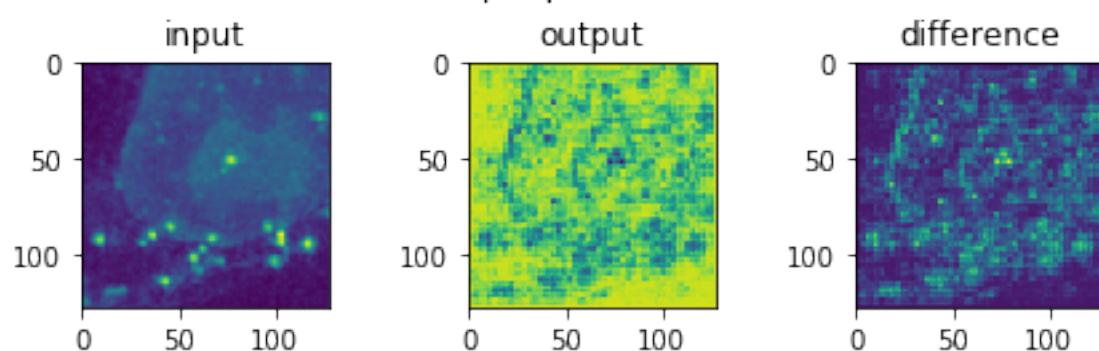


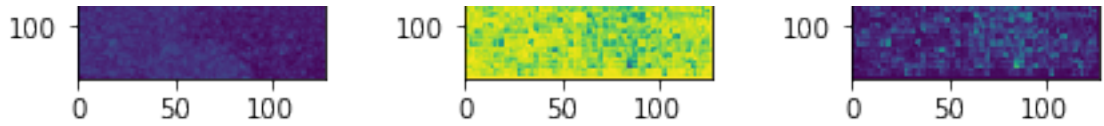


GPS: size_mask=10



UPS, input patch 512*512





○ 분석

	G	GF	GPS	UPS
색반전	X		O	O
격자무늬	X		X	O
Hallucination	Mask_std가 클 때만		O	?

■ UPS

- 어째서인지 색반전이 됨
- Edge에서 difference가 강함
- 격자무늬 artifact가 심함

■ Gaussian

- UPS와 달리, 격자무늬 artifact가 나타나지 않음. 원본의 형태
- 배경 부분은 많이 변화하고, 세포 부분은 적게 변화함
- 원본의 표준변차는 0.07
 - ◆ Mask_std 0.03: 제일 좋음
 - ◆ Mask_std 0.07: 변화가 적음
 - ◆ Mask_std 0.14:
 - ◇ hallucination?
 - ◇ 세포와 배경을 구분하는 능력이 떨어짐 (cmap에서)

■ GPS

- Hallucination
 - ◆ 도넛 모양
- Size_mask가 커질수록 artifact도 커짐

가 보존됨

서 outlier가 생겼기 때문일수도 있음!)