Monday, December 23, 2019

12:50 PM

질문 1. 3D-N2V가 실패하는 이유는 정확히 무엇인가?

* 사용한 코드: N2V 저자가 제공한 tensorflow implementation <https://github.com/juglab/n2v>
* 프로젝트 위치: 0번서버 /home/user/jaeun/n2v/
* 저자들이 제공한 작동 예시: 3D 초파리 날개

A picture containing monitor, object, screen

Description automatically generated

* 관찰한 점:
  + blur현상이 조금 일어나긴 했지만 날개의 그물 구조는 여전히 보존된다.
  + 배경이 잘 denoise되었다.
  + output intensity가 전체적으로 낮아졌다.
    - 3D flywing output histo 
      Input 
      output 
      1000000 
      800000 
      600000 
      400000 
      200000 

* 3D artificial bead에 적용한 결과:
  + N2V가 대체 픽셀을 끌어올 수 있는 범위 ('neighborhood\_radius')값을 다양하게 바꿔보았다.

3D-N2V with Various Neighborhood Radius 
100 
120 
Input. Middle slice w_r.t Z 
8) 100 120 
Neighborhood Radius 3. Middle slice Z 
Diff input (ignore) 
8) 100 
Neighborhood Radius 3, Diff 
100 
120 
20 40 
100 120 
100 
120 
100 
120 
20 40 
8) 100 
Neighborhood Radius 5, Middle slice w.r_t Z 
100 
120 
80 
100 
120 
20 
Neighborhood Radius 5, Diff 
100 
40000 
120 
630000 
500000 
400000 
300000 
200000 
100000 
120 
140000 
120000 
100000 
40000 
120 

* 관찰한 점: output intensity range가 제멋대로 변한다.
  + neighborhood radius가 너무 작으면 (3) 결과값이 하나로 통일되어 background와 signal을 구분할 수 없다.
  + neighborhood radius가 5일 때 background와 signal을 구분할 수 있지만 halo effect가 심하다.
  + neighborhood radius가 약간 커져도 (7) 결과값은 neighborhood radius가 5일 때와 크게 달라지지 않는다. 다만 bead 경계에 마이너스 값을 가진 픽셀들이 생겨 halo effect가 조금 악화된다.

* 3D N2V에서 neighborhood radius가 너무 작을 때 (3) 값이 하나로 밀리는 문제가 나타났으므로,  
  2D N2V에서도 neighborhood radius를 작게 한 실험의 결과:

2D-N2V With Various Neighborhood Radius 
1000 
o 
1000 
o 
1000 
o 
Input 
200 400 600 800 
Neighborhood Radius 3 
200 400600 800 
Neighborhood Radius 5 
200 400 
Neighborhood Radius 7 
1000 
1000 
1000 
200 
400 
800 
o 
200 
400 
o 
200 
400 
600 
o 
200 
400 
600 
Diff input (ignore) 
200 400 600 800 1000 
Neighborhood Radius 3, Diff 
200 400 600 800 1000 
Neighborhood Radius 5, Diff 
200 400 1000 
Neighborhood Radius 7, Diff 
70000 
60000 
50000 
40000 
30000 
20000 
10000 
200000 
150000 
100000 
50000 
400000 
300000 
200000 
100000 
200000 
175000 
150000 
125000 
100000 
75000 
50000 

* 관찰한 점: 2D N2V에서는 neighborhood radius가 결과의 질을 크게 좌우하지 않는다.
  + Buades 논문에 의하면 차영상이 랜덤 노이즈와 가장 비슷한 방법이 가장 훌륭한 디노이징 방법이다. 그런 면에서 차영상에 동그라미가 보이지 않는 neighborhood radius 5 경우가 가장 성능이 좋다.
    - neighborhood radius=5인 경우의 히스토그램이 가장 폭이 좁다. 이것은 noise range가 좁다는 뜻이다.
  + 그러나 최적 파라미터 (neighborhood radius=5)에서 벗어난 경우에도 halo가 아주 조금 생기는 것 빼고 큰 차이가 없다.
* 아직 모르는 것:
  + 왜 3D-N2V 는 neighborhood radius라는 파라미터에 큰 영향을 받는데 2D-N2V는 그렇지 않은가?

질문 2. DIP가 실패하는 이유는 정확히 무엇인가?

* 사용한 코드: DIP 저자가 제공한 pytorch implementation <https://github.com/DmitryUlyanov/deep-image-prior>
* 프로젝트 위치: 0번서버 /home/user/jaeun/deep-image-prior/

* 문제 상황: 2D artificial bead에 대한 DIP
  + artifact가 생겼다
  + signal(노란 동그라미)의 경계선이 구불구불

**DIP on 2D artificial bead, single channel**

* 0001 08 0 0 OOZ 
  OOZ 
  00 

* 실험 1: 저자가 제공한 예시는 3D RGB 이미지에 대한 것인데 우리 artificial bead는 single channel image라서 차이가 생기나? Artificial bead를 3 channel로 바꿔서 해보기.

2D DIP on 3-channel Image 
noisy input 
Output. channel 1 
200 
EOO 
Output, channel 2 
200 
400 

* 실험 2: artificial bead의 경계선이 step function이어서 image prior가 작동하기 어려운가? 부드러운 경계선을 가진 sinusoidal artificial bead에 대해 DIP를 적용해보기.

Noisy Input, wavenumber 1 
1000 
o 
GT, wavenumber 1 
250 500 750 
GT. wavenumber 5 
1000 
1000 0 
250 500 750 1000 
Noisy Input. wavenumber 5 
200 
400 
€00 
800 
200 
400 
DIP 
DIP 
200 
400 