

Project

▼ 이미지 소개

1. Denoising: Heavy함

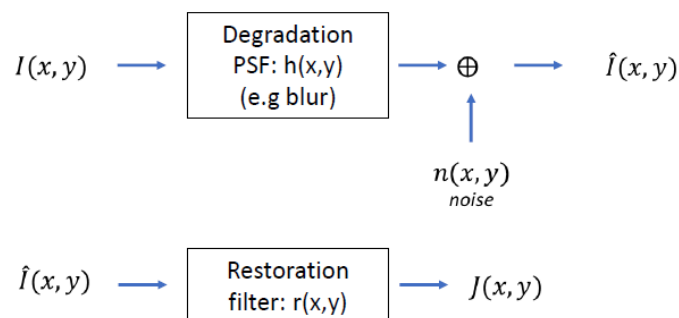
- Histogram으로 noise의 종류를 확인한다.

2. Noise가 아닌게 있는 경우는....? Noise와 아닌 영역을 잘 구분해야 함.

3. ?? → Image Degradation → **Image Restoration** → "Wiener filter", "Deep Learning", ...

▼ 풀이 과정

Basic model for degradation



$$\hat{I}(x, y) = I(x, y) * h(x, y) + n(x, y) \quad J(x, y) = \hat{I}(x, y) * r(x, y) + n(x, y) * r(x, y)$$
$$\hat{I}(u, v) = I(u, v)H(u, v) + N(u, v) \quad J(u, v) = \hat{I}(u, v)R(u, v) + N(u, v)R(u, v)$$

Degradation은 LSI system으로 가정, Convolution을 거쳐 만들어진다고 본다.

Degradation에 대한 수학적 식은 제공할 것임. \Rightarrow 주어진 forward 모델은 알고있으니, 복원하는 걸 만드는 게 Challenge.

단 Degradation을 해결하기 힘든 것을 제공할 예정임.

즉 h 는 주어지고, n 은 알아내야 함. r 을 설계해야 함

결국, Denosing과 Restoration은 Deep Learning 기반을 푸는 게 가장 이미지 복원이 잘 될 것임.

[Section A: Denoising]

▼ Noise 확인하기

이미지마다 Noise Characteristic이 다른지, Noise Power가 다른지 분류를 해야한다.

최소 70장 내지는 700장은 직접 보고 분석해보기를 추천함.

▼ Denoising

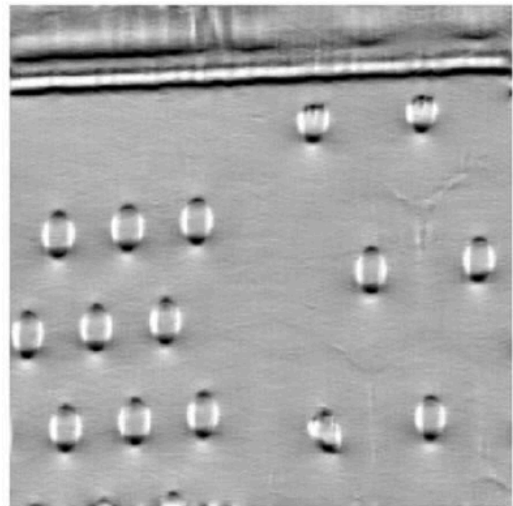
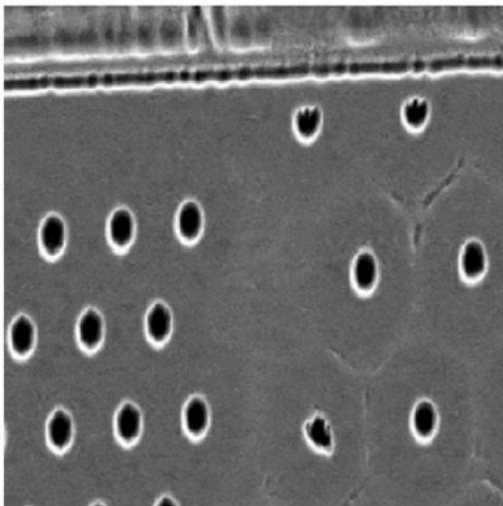
- LPF
- Median filter
- Adaptive filter (강력하진 않는데, 우리 영상은 꽤 단순해서, 은근히 잘 통할 것임)
- Deep learning 기반 denoising

[Section B: Restoration]

▼ Kernel Design

Degradation 이미지를 보면 Gaussian처럼 일정하지 않고, 상하 방향과 좌우 방향이 다르다. ⇒ 어떻게 Design할까?

*Noise 제거 후의 이미지



해결하기 위해서는 나누는 방법을 사용...? ⇒ Noise가 있을 때에는 비효율적임 (Noise가 Amplify됨)이니 Noise를 제거하고 사용하라.

$$\hat{I}_S(u, v) \quad \text{vs.} \quad \hat{G}_S(u, v)$$

S = guess area Manual guess

$$H(u, v) = \frac{\hat{I}_S(u, v)}{\hat{G}_S(u, v)}$$

그럼 어떻게? Wiener Filter

$$\bar{I}(u, v) = \left[\frac{1}{H(u, v) |H(u, v)|^2 + K} \right] \hat{I}(u, v)$$

↑
Tunable parameter

이러한 Classic한 방법 vs. Deep Learning 방법으로 풀어보라.

[Idea: Neural Network 사용 시?]

Section A, B가 모두 Neural Network를 쓴다면, 둘을 통합하는 방법도 가능할 것임.
Simple한 Neural Net을 사용하면 된다.

test_x를 만들어서, 거기에 넣은 파일들을 가지고, metric에 넣어서 평가를 하게 된다.

이미지는 256*256, 이미 0~1로 Normalize된 이미지이다.

방향이 다른 5개씩의 이미지가 나올 것임 ⇒ 교수님의 추천은 모든 방향의 Average를 내볼 것임.

Argument가 중요하다. 왜 어떤 방식을 썼는지. 이 부분이 정성적인 평가에 포함될 예정.

그리고 다섯개의 orientation은 CT 방식으로... ⇒ Classic한 방법으로 풀 때 장점은?

Deep learning은 consistency가 깨지면 틀릴 수 있음. Classic한 방법으로 풀면 consistency가 깨지더라도, 각도만 수정해주면 명확히 풀 수 있다.

반대로 한 장의 이미지, 혹은 소수의 이미지만으로 풀 기 위해 Deep Learning을 사용했다는 의도로 표현해도 OK

혹은 현업에서는 Label이 없을 수 있으므로, 비지도 학습으로 진행해도 OK.

Degradation에 대한 Logic은 주어진 예정.

Noise는 주어지지 않음. 직접 Noise의 유형과 세기를 파악 후 Denosing 해야 함.

Noising 자체에 대한 함수는 제공은 됨.

[Tip]

Diffusion Network를 사용하면 Denoising을 되게 잘해줌

[1단계: 이미지 분석]

Noise를 어떻게 판단할 것인가 (Histogram)

Degradation을 어떻게 판단할 것인가

[2단계: Denoising]

방법론을 고민 필요

[3단계: Restoration]

방법론을 고민 필요

[코드 설명]

code_denoising 안의 run_test.ipynb는 같은 폴더 내의 test.py를 가지고 돌리는데,

그 안의 class ModelType을 수정하면, filter 폴더 내의 mean_filter.py와 median_filter.py를 골라 적용 가능함. (어케 쓰라는겨...)

train.py 내의 실제 폴더명은 알아서 수정해야 함.