

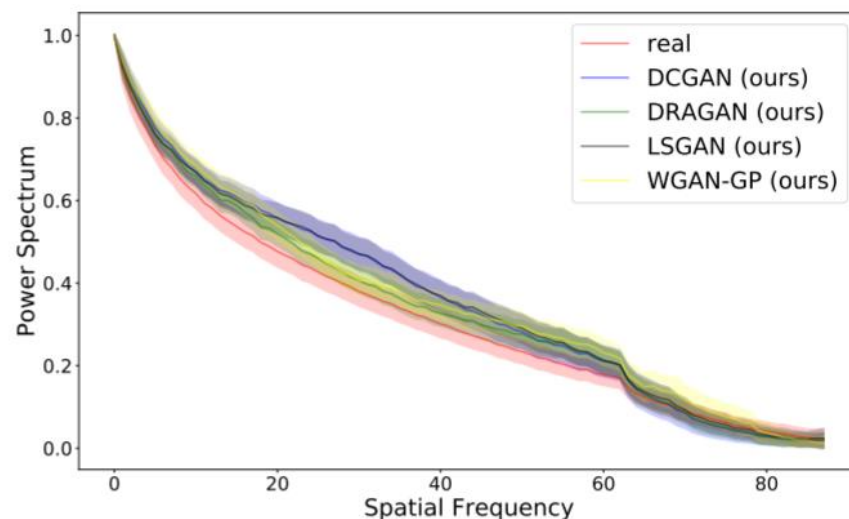
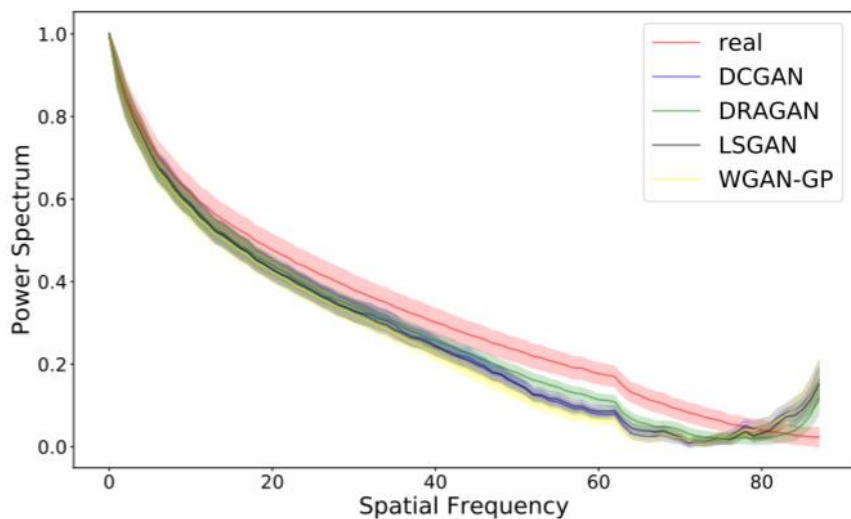
Watch your Up-Convolution

을 통해 논문 작성에 대한 아이디어 이해하기

20185153 이상영

• 논문의 Idea

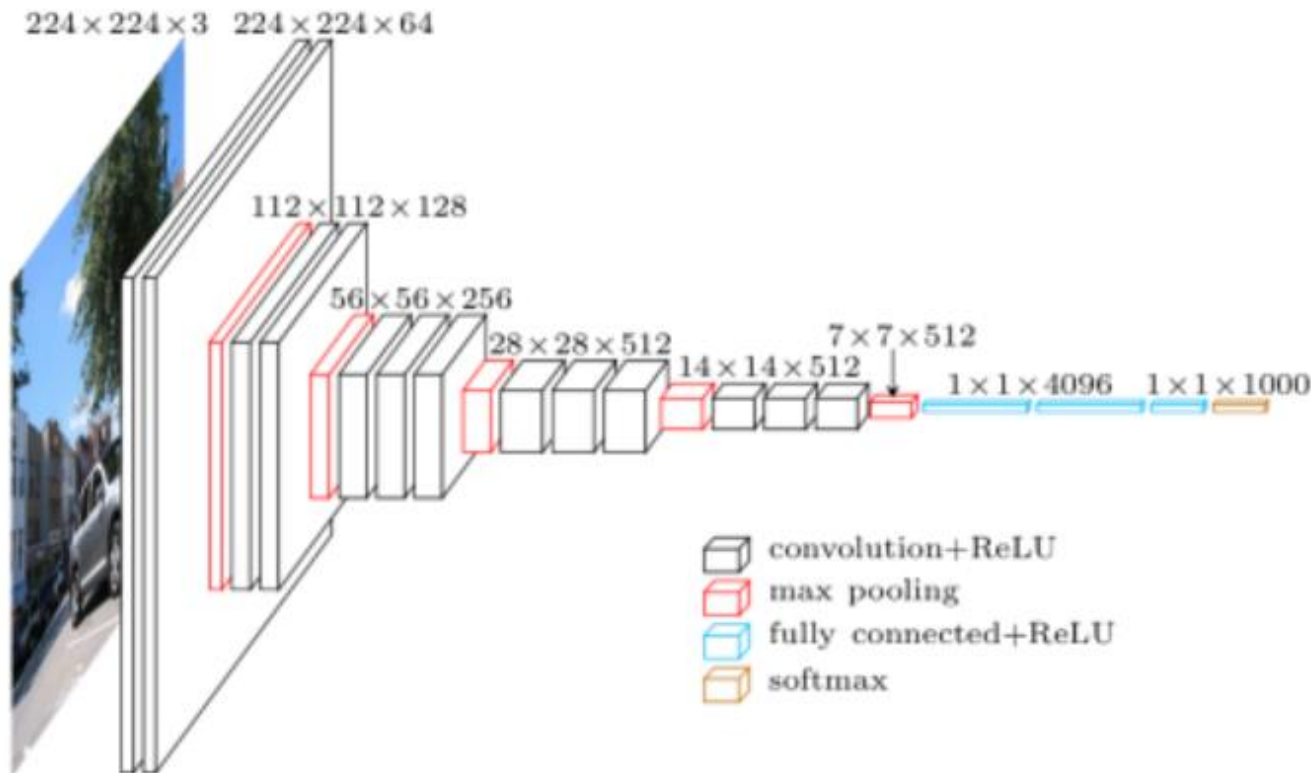
- GAN, AE와 같은 생성적 신경망; **Up-Convolution** 기법을 사용
 1. 보간을 이용한 up-conv
 2. 전치된 conv
- **Up-Conv이 스펙트럼 왜곡을 유발한다는 사실 발견** → 딥페이크 검출기 제안
- 스펙트럼 왜곡을 보정할 수 있는 Spectral Regularization 제안
→ GAN 모델 안정화 & 시각적 출력 품질 향상



• 키워드

1. Up-sampling method(up-conv/transposed-conv)
2. 주파수 영역
3. 이산 푸리에 변환
4. 방위각 적분

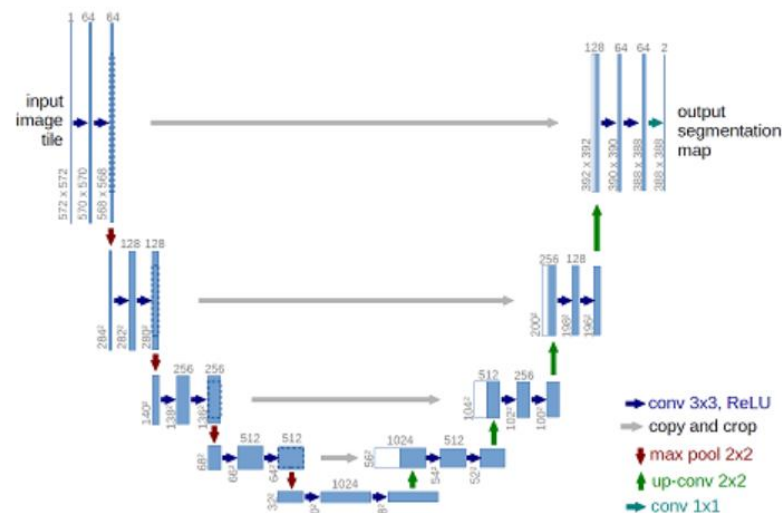
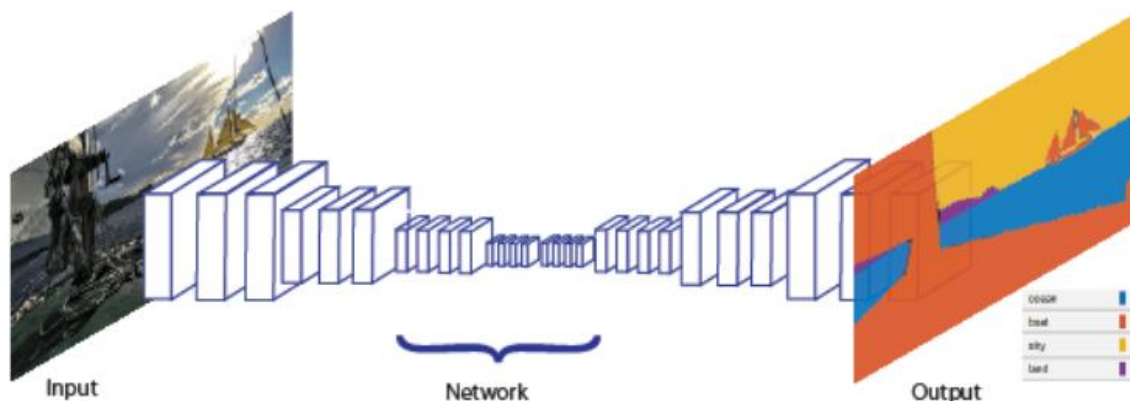
- **Up-sampling method**(up-conv/transpose-conv)



- **Down sampling**

- 보통의 conv 연산; output의 크기가 input보다 작아지는 특성 존재
 - Conv layer가 깊어질 수록 \rightarrow feature map의 크기 작아짐

• Up-sampling method(up-conv/transpose-conv)



- Up sampling

- Output의 크기가 input보다 커지게하는 방법
 1. Unpooling
 2. Transposed conv

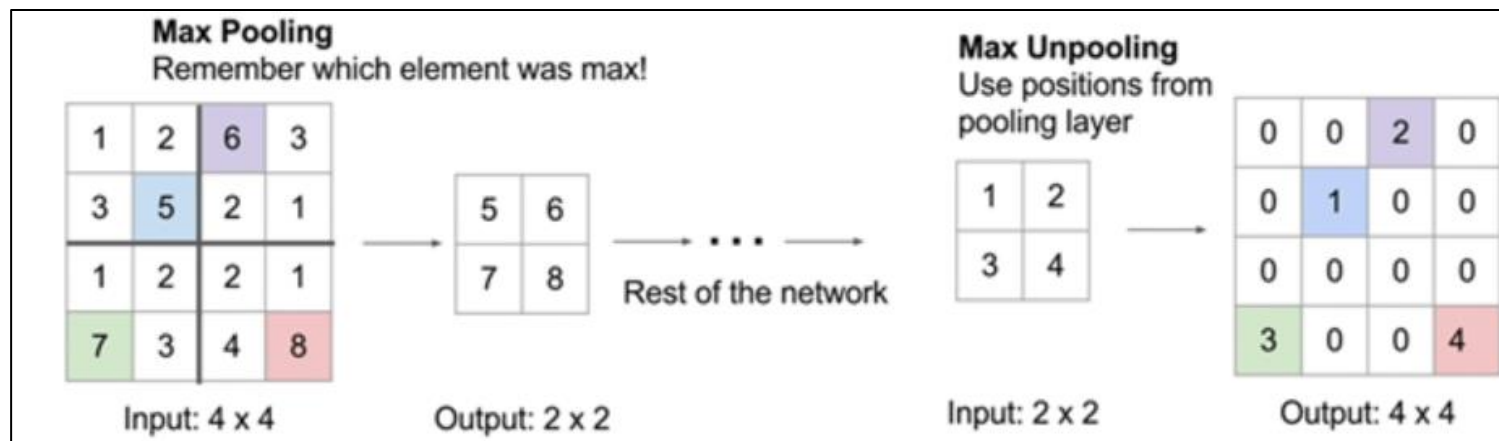
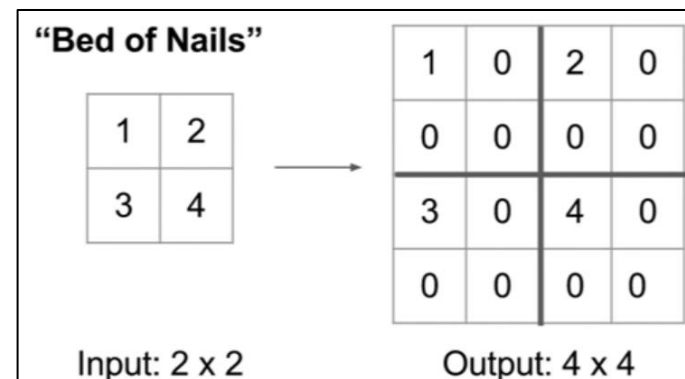
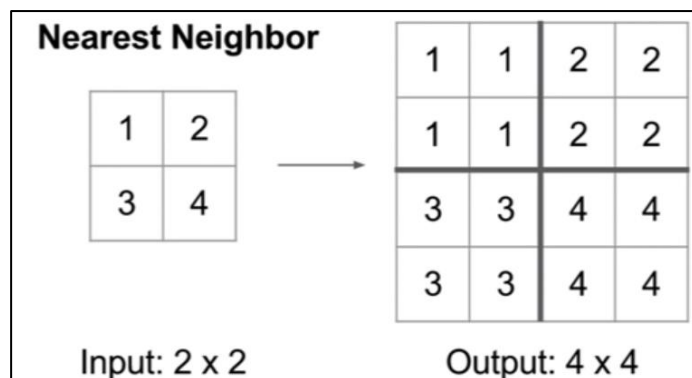
- Up sampling 사용 예시

- Semantic segmentation
 - Input 이미지를 압축된 벡터로 표현 → 원래의 input 이미지와 동일한 크기로 되돌림.
 - 이미지의 픽셀단위로 예측하기 위해 높은 해상도 필요
- U-net
 - 모델의 layer마다 얻어지는 feature map을 합치는 구조
 - 서로 모양이 맞아야 feature map을 합칠 수 있음.

• Up-sampling method(up-conv/transpose-conv)

- Unpooling

- Pooling layer를 복원하는 방법

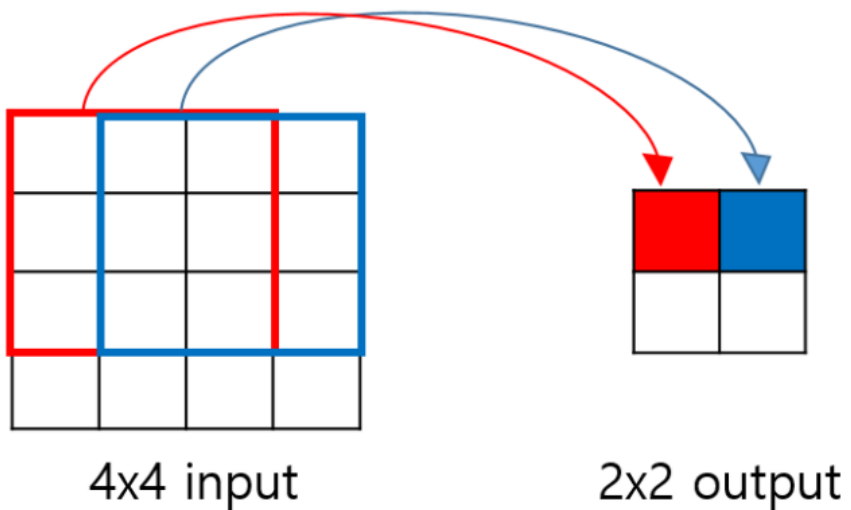


• Up-sampling method(up-conv/transpose-conv)

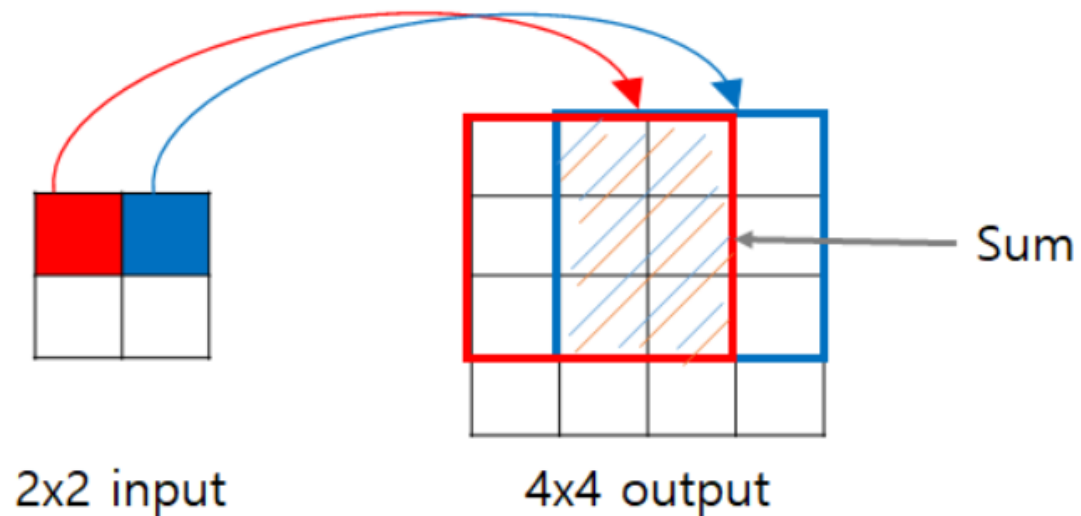
- Transposed conv

- Stride로 인해 발생하는 down sampling에 대한 복원

3x3 Convolution, stride 1, padding 0



Transposed Convolution



• Up-sampling method(up-conv/transpose-conv)

$$f * g = h$$

- f: filter
- *: convolution 연산
- g: input
- h: output

- Deconvolution

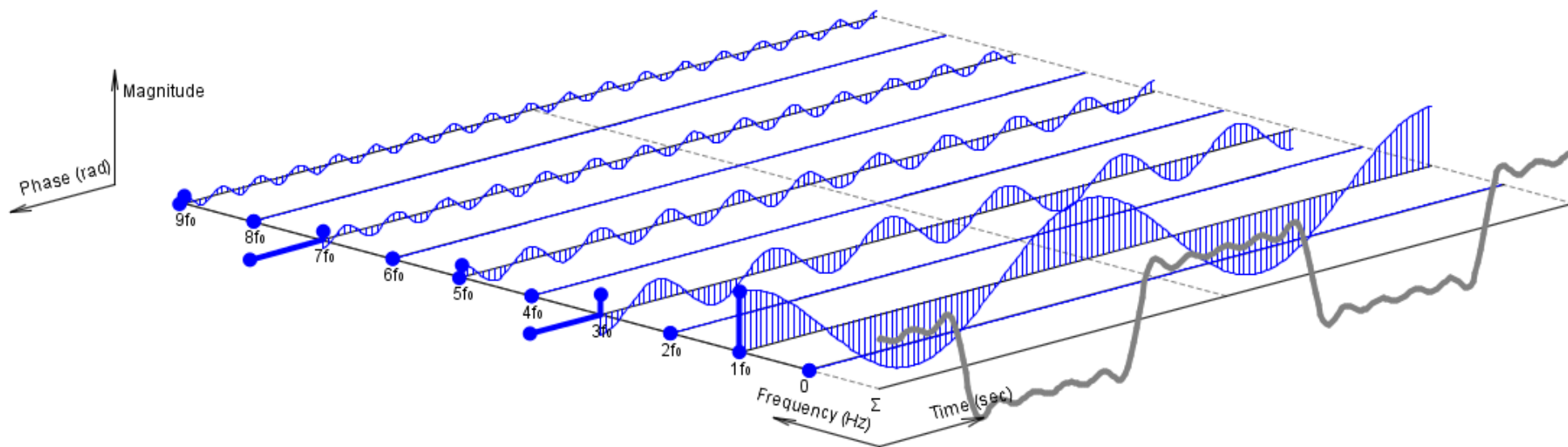
- Convolution의 역 연산
- f와 h를 알고있는 상태에서 g를 구하는 것
- Conv layer를 통과시켜서 얻은 feature map → de-conv → input

- Transposed-Conv Vs. De-conv

- Transposed-Conv: 여기에서 사용하는 kernel은 어떤 conv layer와 공유하는 것이 아니라, 학습을 통해 찾아감.
- De-conv: 사용한 kernel과 output을 알고 있고, 역 연산을 통해 input을 재현하는 것이 목적

• 주파수 영역

- 주파수 단위로 정의되어있는 \cos , \sin 파동함수 \rightarrow 이미지 신호를 해석하기 위해 기저함수로 사용
- 파동신호는 서로 독립적(직교) \rightarrow "선형결합"을 통해 모든 신호를 표현할 수 있음



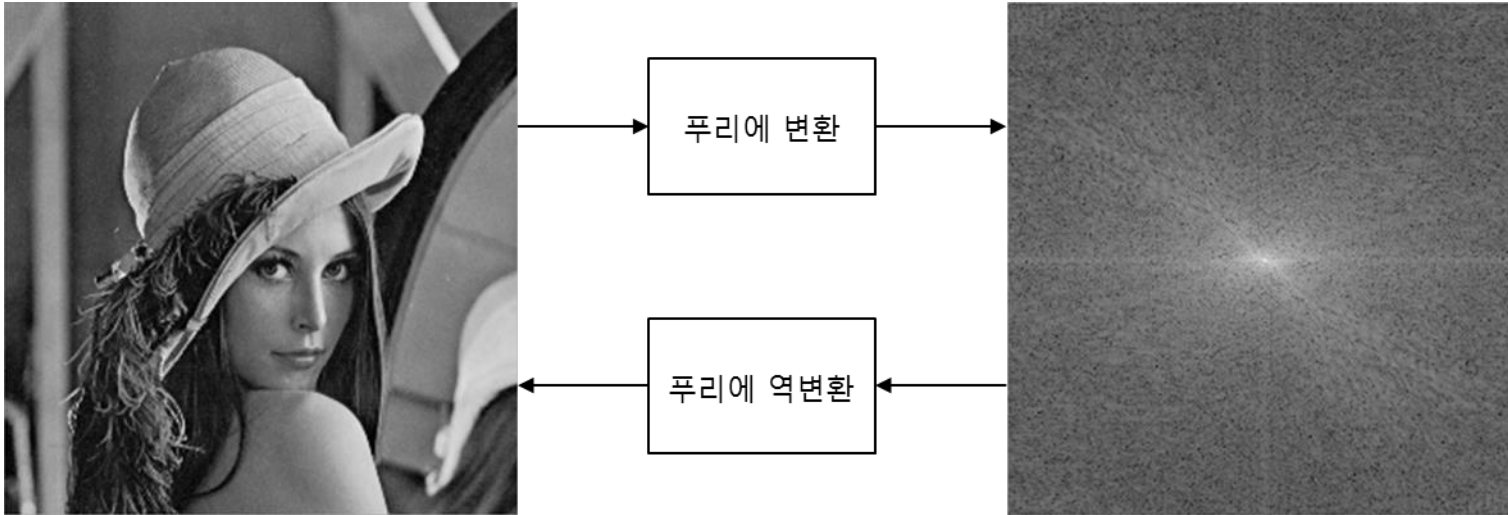
• 이산 푸리에 변환

- 이산 푸리에 변환(Discrete Fourier Transform)
 - 임의의 입력신호를 다양한 주파수를 갖는 주기함수들의 합으로 변환하는 것

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$$

$$j = \sqrt{-1}, \quad e^{\pm jx} = \cos(x) \pm j \sin(x)$$

• 이산 푸리에 변환



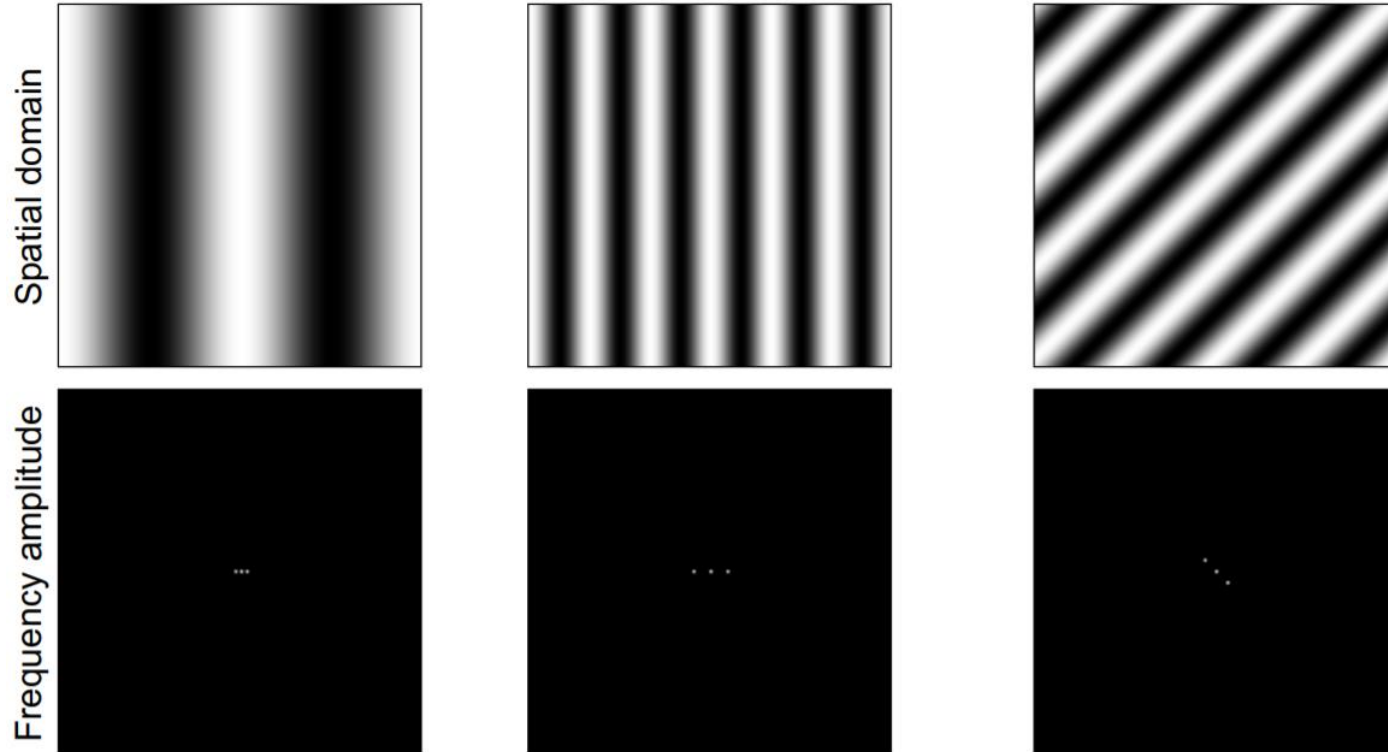
공간영역

: 픽셀로 구성된 일반적인 이미지 영역

주파수 영역

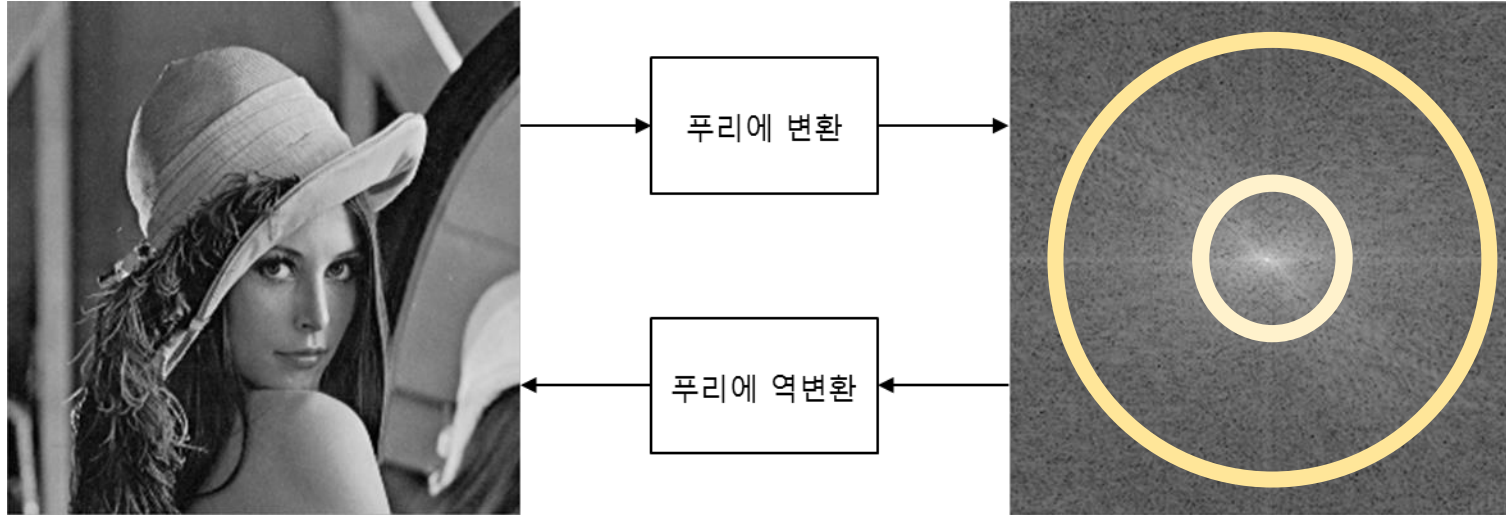
- 푸리에 스펙트럼: 해당 주파수 성분이 이미지에 나타나는 강도 표현

• 이산 푸리에 변환



- Spatial domain(공간)
- Frequency amplitude(진동수의 진폭)
 - Frequency가 어느정도의 빈도로 나타나는지 표현
 - 중심으로부터의 거리
 - 가까울수록; 진동수, 빈도수 ↓ (저주파)
 - 멀수록; 진동수, 빈도수 ↑ (고주파)

• 이산 푸리에 변환

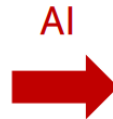
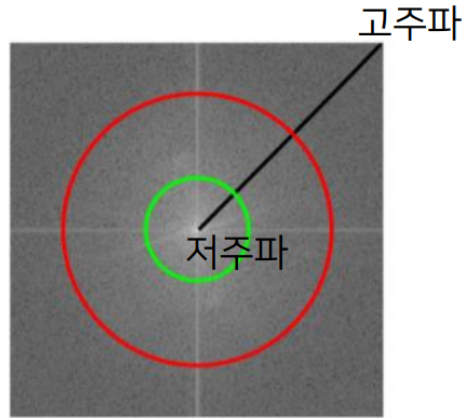


공간의 반복수에 대해

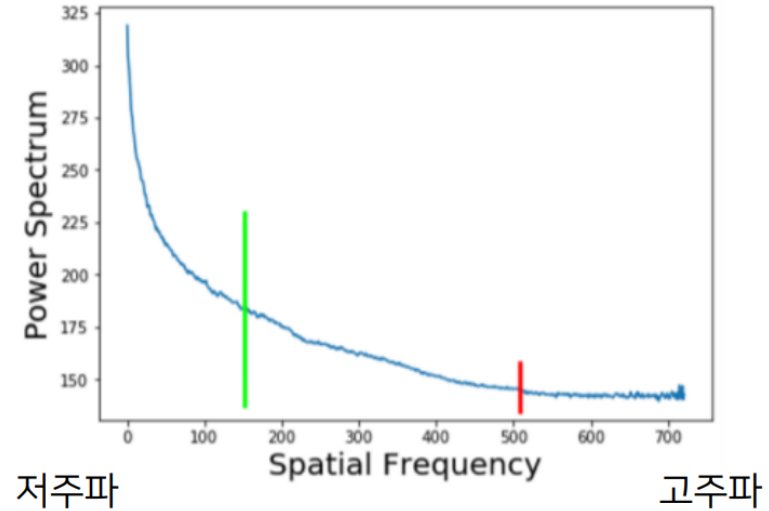
多	high pass	고주파	날카롭고 변화 큰 부분
少	low pass	저주파	blur

• 방위각 적분

[2D Power Spectrum]



[1D Power Spectrum]



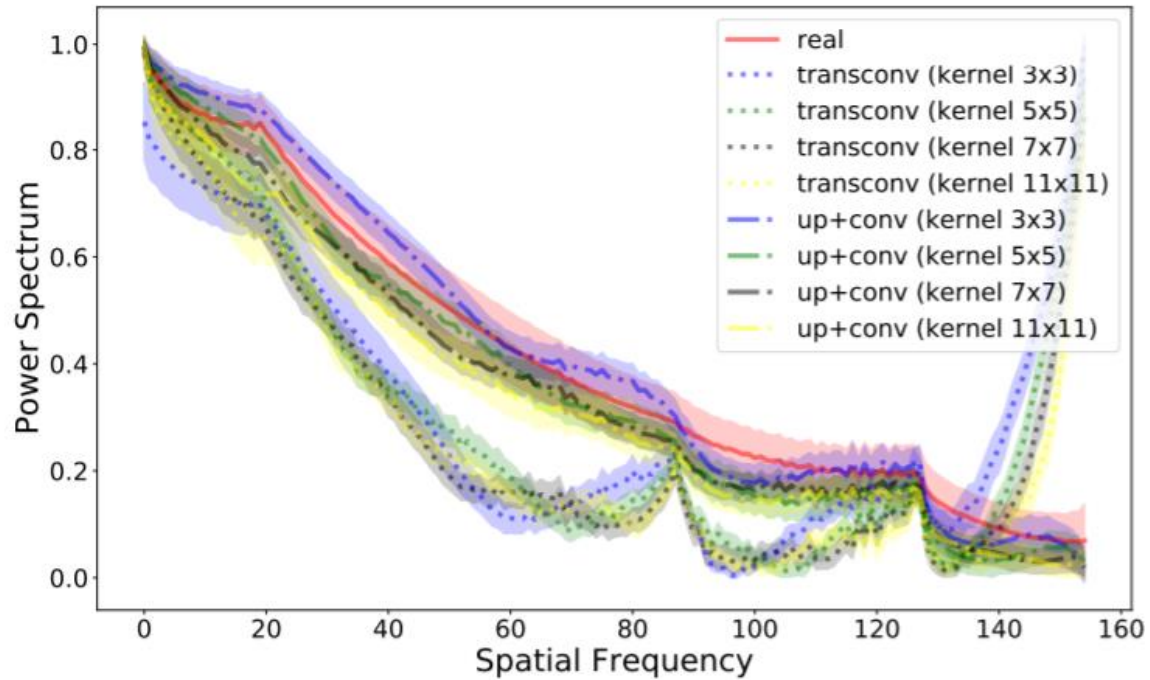
$$AI(\omega_k) = \int_0^{2\pi} \|\mathcal{F}(I)(\omega_k \cdot \cos(\phi), \omega_k \cdot \sin(\phi))\|^2 d\phi$$

for $k = 0, \dots, M/2 - 1,$

: MXM인 스펙트럼 이미지를 입력

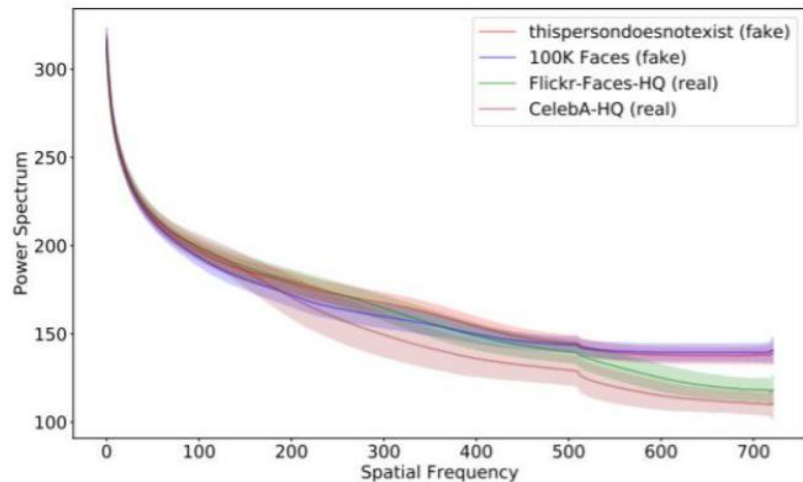
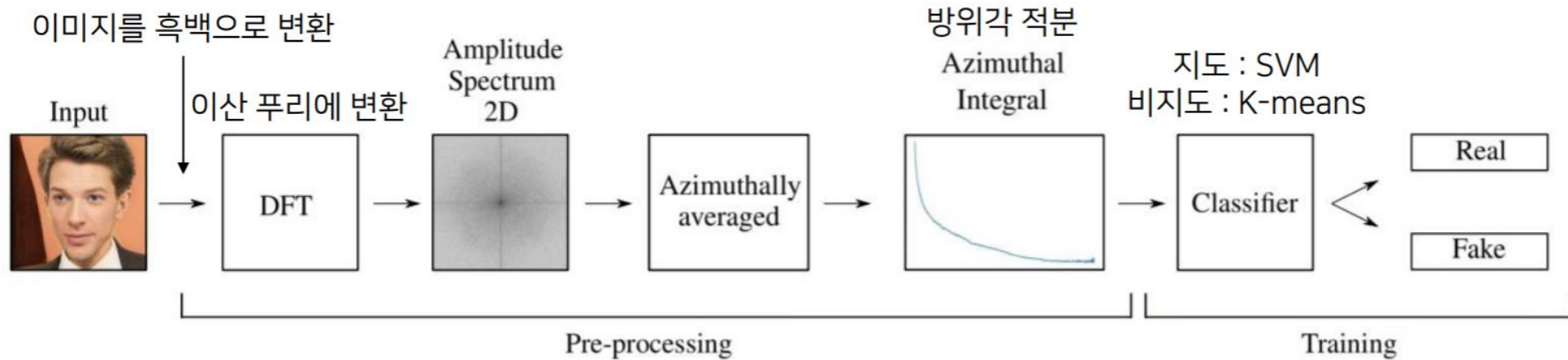
- 반지름 기준으로 모든 성분들의 합과 평균 → 1D Power Spectrum
- X축: 반지름 길이, 주파수 크기
- Y축: 적분값

• 논문- 실험결과



Up+conv : 고주파 결여 → blurring 효과
Transconv: 고주파 노이즈 多 → 경계부분에 많이 분포

• 논문- 스펙트럼 왜곡을 통한 Deepfake Detection



명확히 real / fake 구분
→ Detection 성공!

data set	80% (train) - 20% (test)			
	method	# samples	supervised	unsupervised
Faces-HQ	ours	1000	100%	82%
Faces-HQ	ours	100	100%	81%
Faces-HQ	ours	20	100%	75%
CelebA	ours	2000	100%	96%
CelebA	[57]	100000	99.43%	-
CelebA	[39]	100000	86.61%	-
FaceForensics++	ours ^A	2000	85%	-
FaceForensics++	ours ^B	2000	90%	-

감사합니다.