# WGAN-GP를 활용한
# MRI 뇌 영상 이미지 Super Resolution 수행

서범진

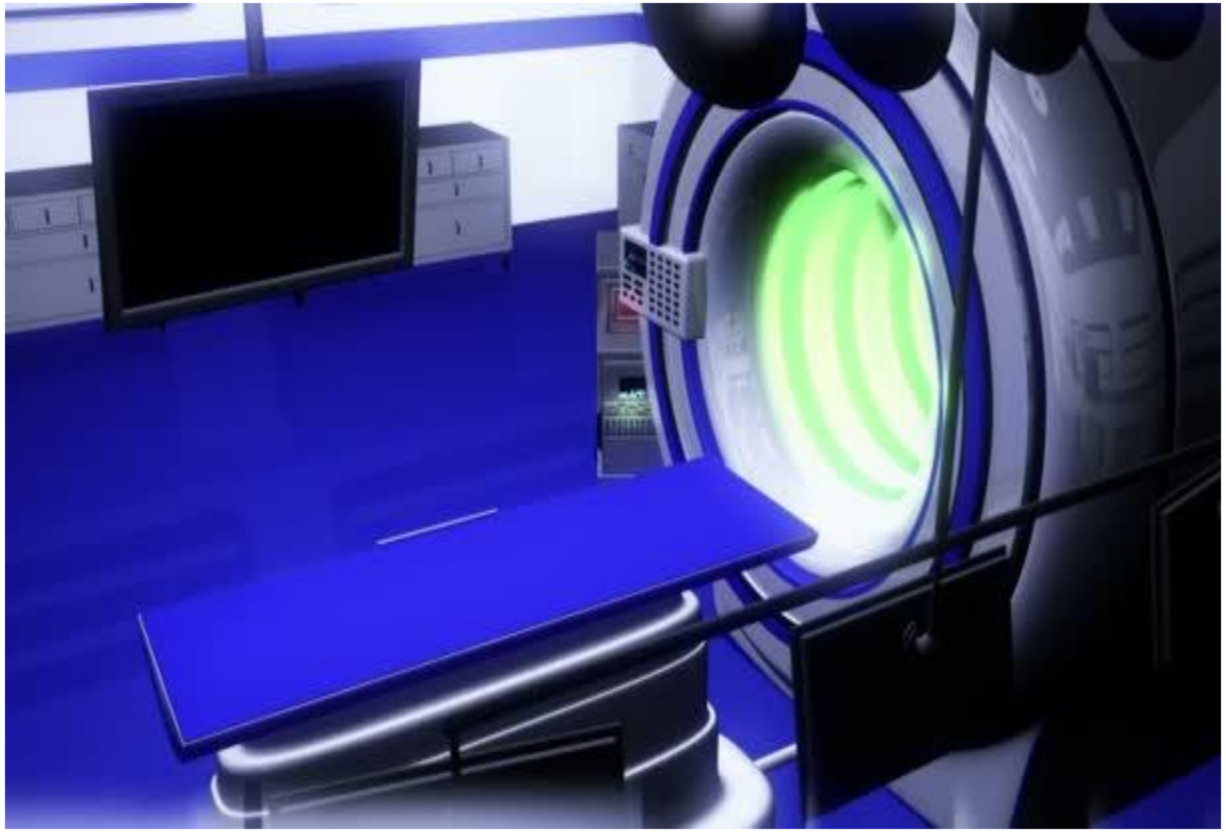
# 연구 주제

- 배경 및 문제점(HR MRI)

  - 스캔 시간이 길
  - spatial coverage가 작음
  - signal-to-noise ratio (SNR)가 작음

- 목표 및 기여점

  - 작은 모델 사이즈로 높은 성능을 낼 수 있는 모델을 고안해보자.

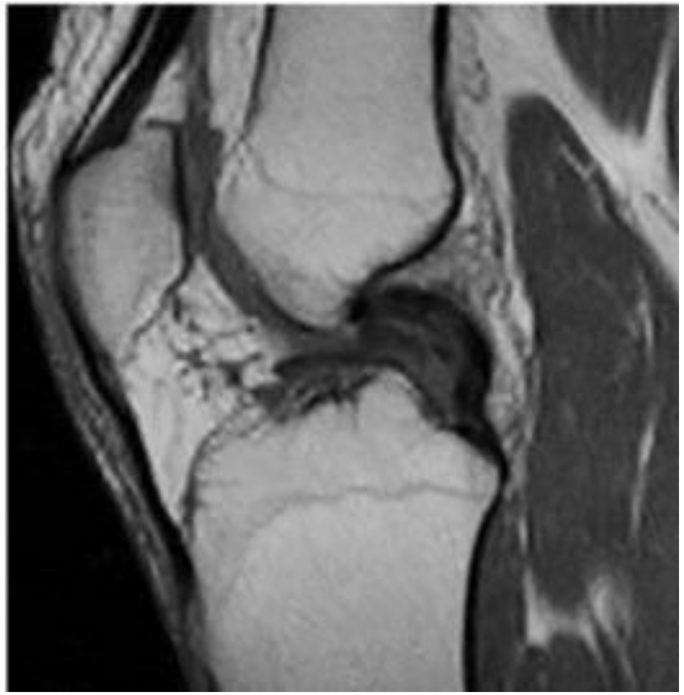


# 관련 연구 _Super Resolution

- Super resolution

low resolution image

High resolution image

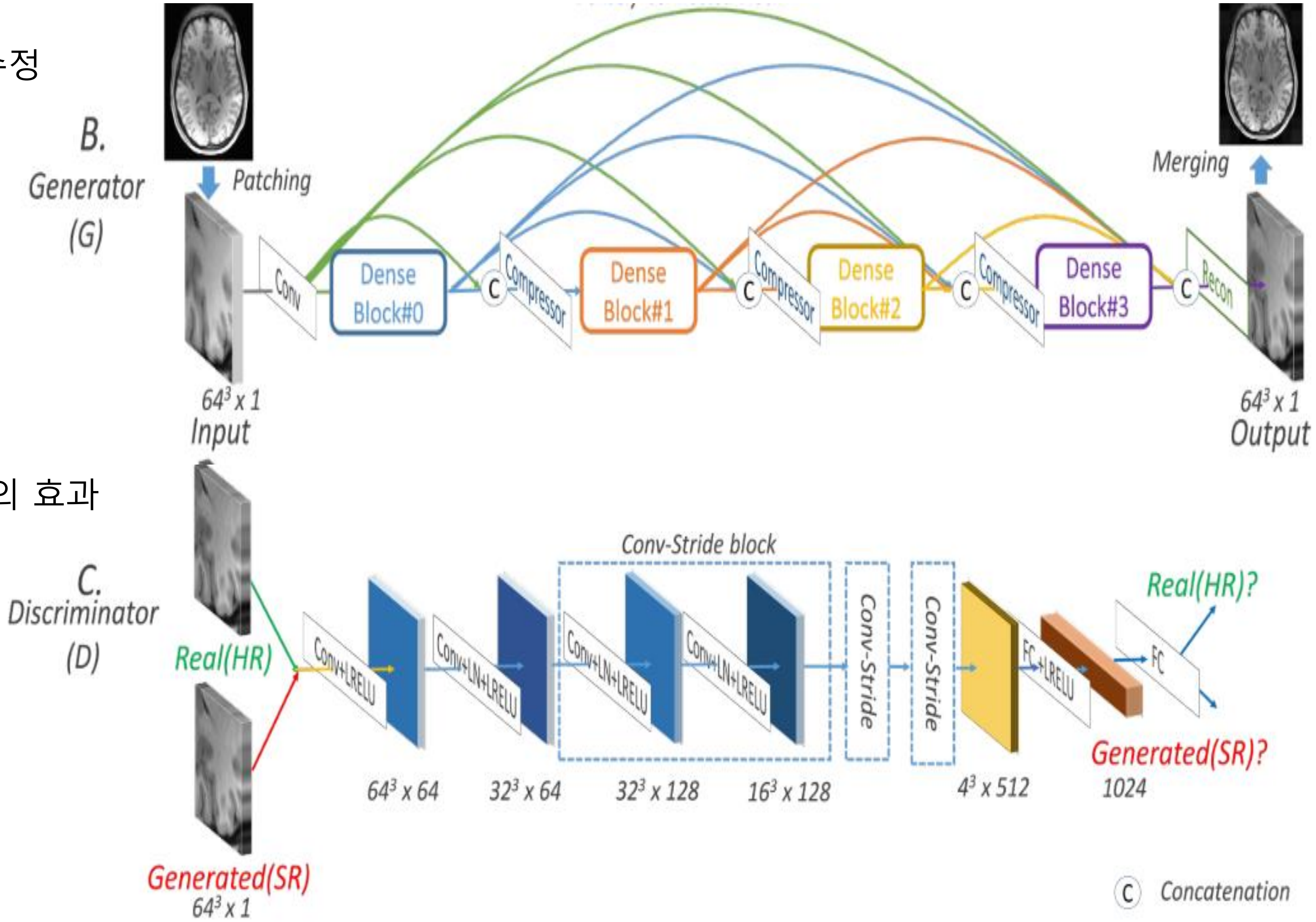


# 관련 연구 _mDCSRN with WGAN-GP

- 기존의 DenseNet을 3D image 에 맞게 수정
- 가장 큰 특징

  모든 채널들이 concatenation 됨.

- 장점
  - Vanishing gradient 개선
  - Feature propagation 강화
  - Feature Reuse, Parameter개수 절약 등의 효과



# 관련 연구 _WGAN with gradient penalty

- Wasserstein GAN
  - 기존의 "분포 사이의 거리"의 metric들의 문제점 해결.

  예를 들어,

  $$KL(\mathbb{P}_r||\mathbb{P}_g) = \int \log\left(\frac{P_r(x)}{P_g(x)}\right) P_r(x) d\mu(x)$$

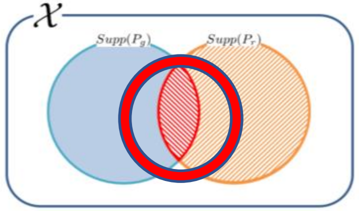
  두 분포의 Support사이의 거리가 멀면 KL이 발산하는 문제 발생.

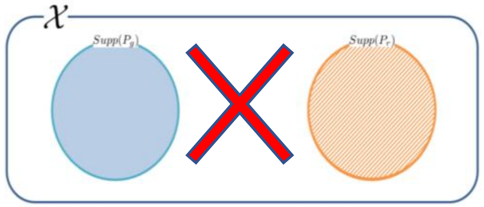
  **Earth Mover distance 개념을 활용한 WGAN 고안**.

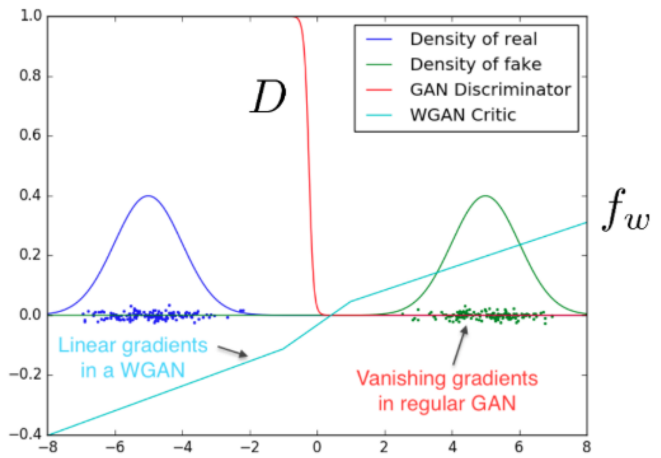
  - model collapsing 문제 해결

  model collasping의 주된 원인은 판별기가 앞서 나가, 판별기의 함수가
  step function으로 수렴, gradient vanishing이 발생

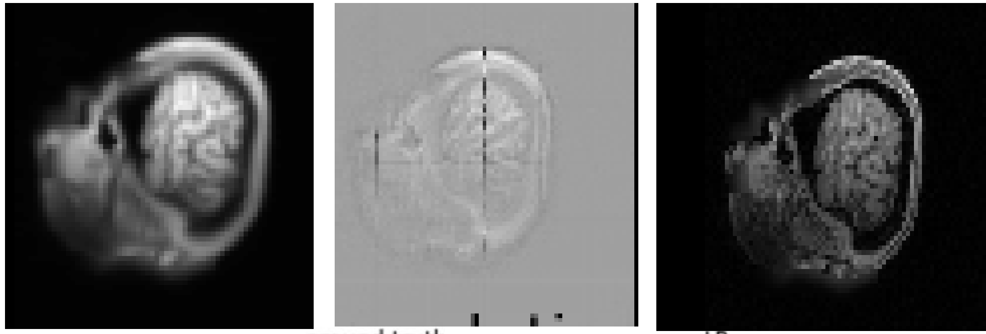
- Gradient Penalty
  - ||f||L≤1라는 제한(Lipschitz constraint)을 만족하기 위해 weight clipping W=[−0.01,0.01]d
  - gradient penalty 의 regularizer term 추가

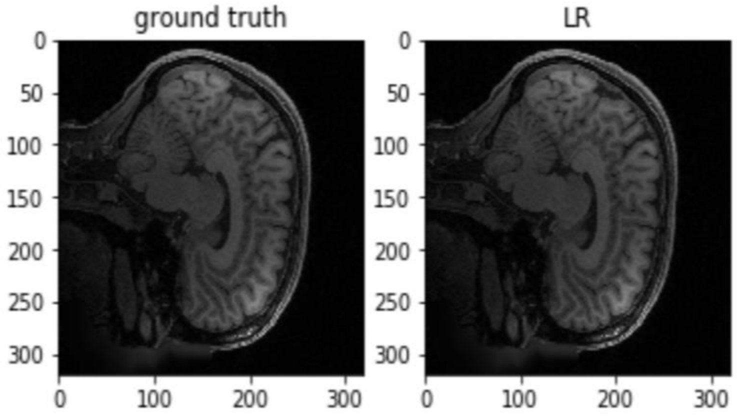


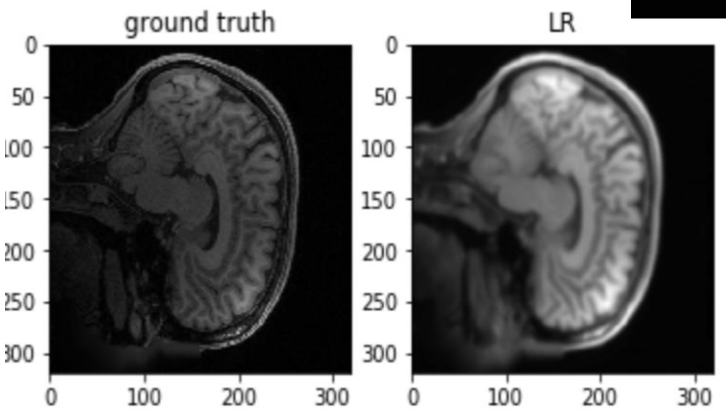



# 제안 방법 _Data Augmentation

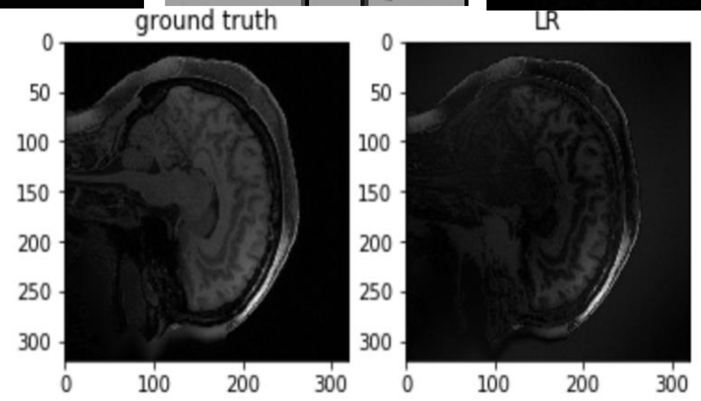




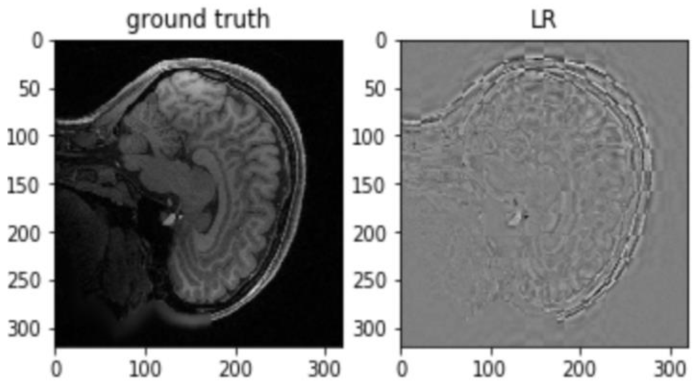

1) A*fft +(1-A)fftshift, zeroing 20



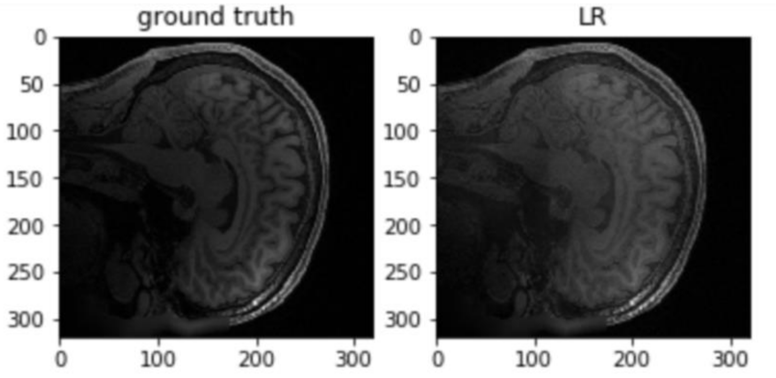

2) fft+Gausian filter, zeroing 20




3) fftn + fftshift

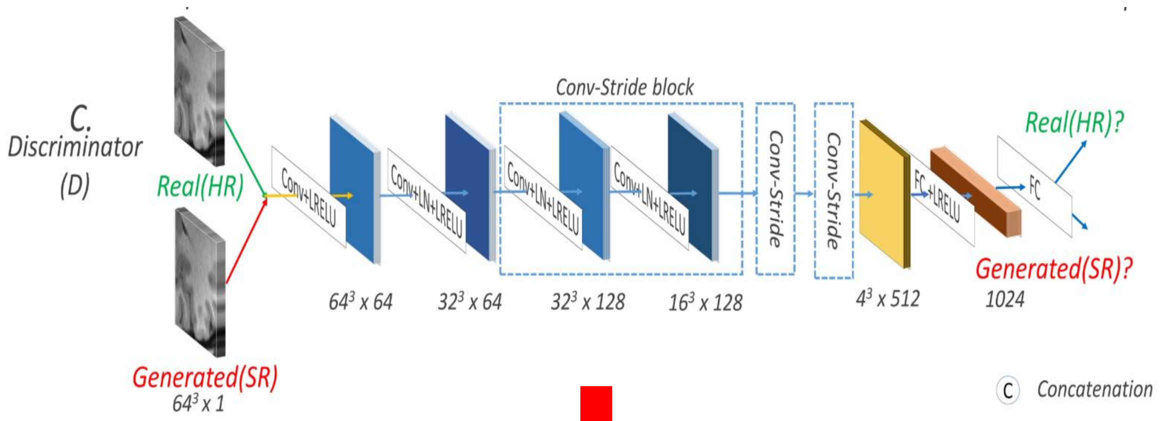


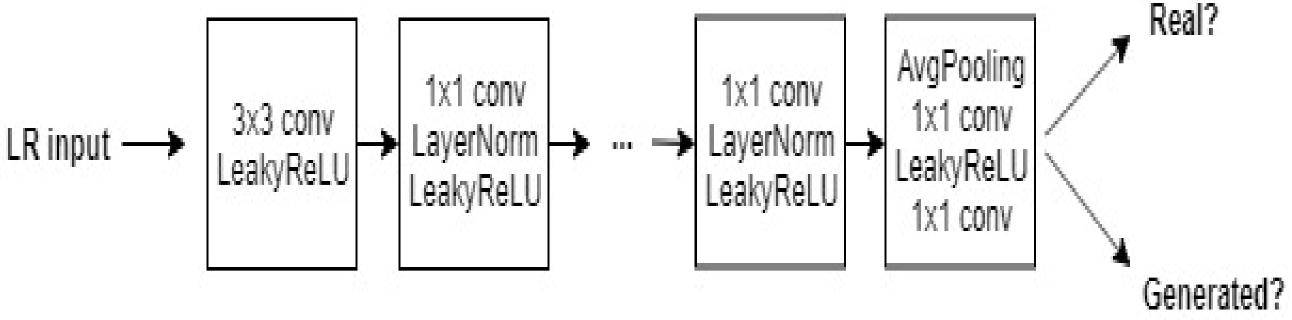

4) fftshift + fftn + fftshift




5) A*fft + (1-A)fftshift, zeroing 160

# 제안 방법 _Model Compression




| | mDSCRN-WGAN | **mDCSRN-WGANGP** |
|---|---|---|
| #param of generator | 0.412M | **0.412M** |
| #param of discriminator | 31.156M → | **17.629M** |

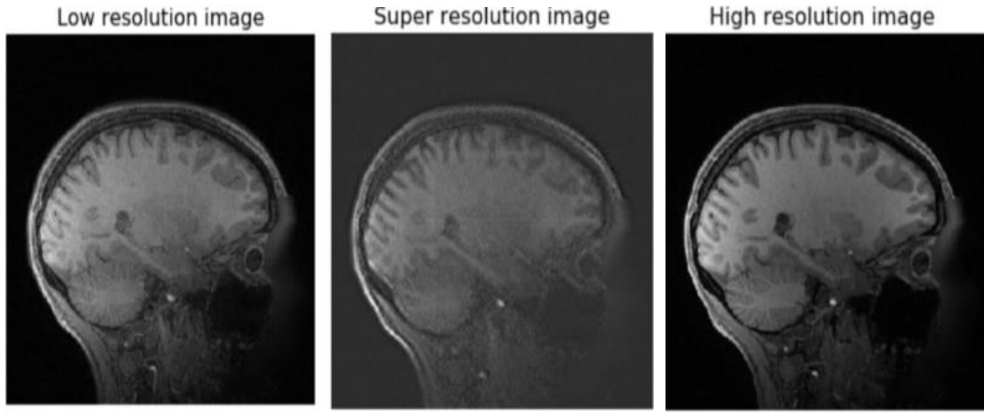
# 제안 방법 _Gradient Penalty term

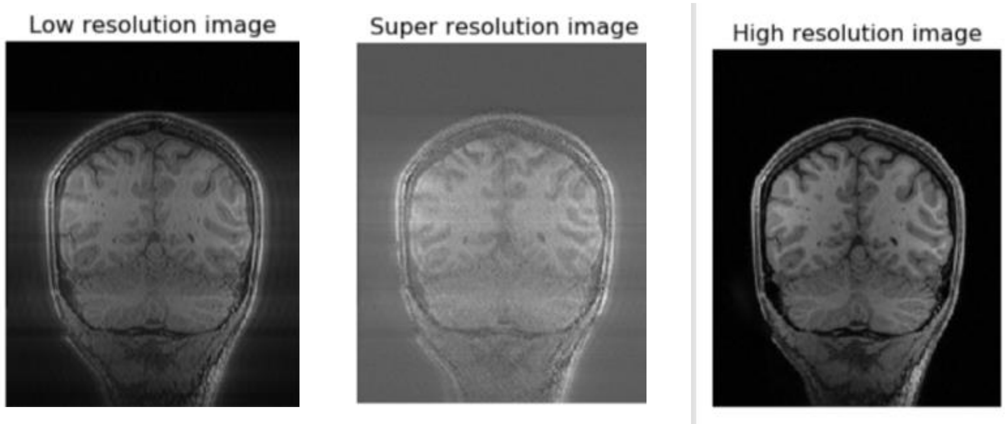
```python
D_loss.backward()
self.optimizerD.step()

# weight clipping
for p in self.netD.parameters():
    p.data.clamp_(-0.01, 0.01)
return sr_patches, D_loss, G_loss, loss
```

```python
# WGAN's Loss
# Calculate Wasserstein Loss
G_loss, D_loss = self.wasserstein_loss(D_fake, D_real)

gradient_penalty = self.calc_gradient_penalty(hr_patches,sr_patches)
gradient_penalty.backward()

D_loss += gradient_penalty

# Semi-supervised Loss (main loss)
#L1_loss from generator and D_loss from Discriminator as wgan-gp
loss = L1_loss + self.lmda * D_loss
```

$$loss = loss_{\text{int}} + \lambda loss_{\text{GAN}}$$

# 검증




| | mDCSRN (250000 step) | | | **mDCSRN-WGANGP (255000 step)** | | |
|---|---|---|---|---|---|---|
| metric | SSIM | PSNR | NRMSE | **SSIM** | **PSNR** | **NRMSE** |
| MEAN | 0.581 | 20.210 | 0.828 | **0.620** | **20.730** | **0.788** |
| STD | 0.133 | 2.496 | 0.155 | **0.123** | **2.838** | **0.181** |

# 결론

- Data Augmentation, model compression, gradient penalty 적용 등 크게 3가지의 model 개선을 수행.
- 최초로 pytorch를 활용해 mDCSRN-WGANGP 구현.
- 중간 결과를 통해 유추해 볼 때, 기존의 model과 비교해봐도 개선된 성능이 예상됨.

감사합니다