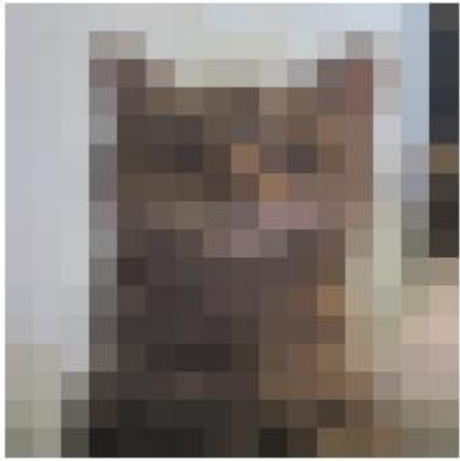


# GLEAN : Generative Latent Bank for Large-Factor Image Super-Resolution

신민성

# Introduction

- SR(Super Resolution) 생성에 새로운 모델을 제안합니다.
  - 손상된 이미지의 디테일한 부분과 질감을 표현할 수 있는 8x to 64x 모델을 제안합니다.
  - 기존에 SR에 사용되는 2가지 접근의 한계를 개선했습니다.
    1. SRGAN, ESRGAN (LR 이미지를 바로 HR로 생성하는 모델)
    2. GAN Inversion (LR이미지에서 latent vector를 생성해 고화질로 생성하는 모델)
  - 학습된 Style GAN을 latent bank 생성모델로 사용했습니다.



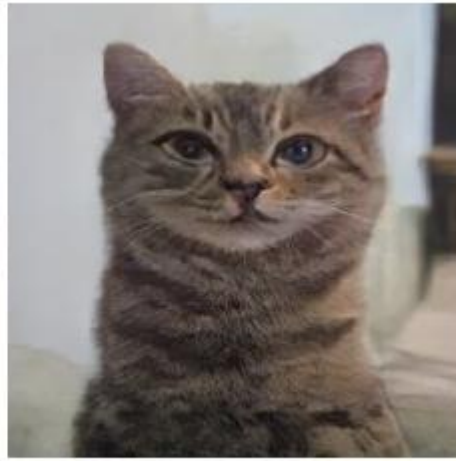
(a) Low-Resolution



(b) ESRGAN



(c) PULSE



(d) GLEAN (ours)



(e) Ground-truth

Example of large-factor super-resolution (16x)

# Related Work

---

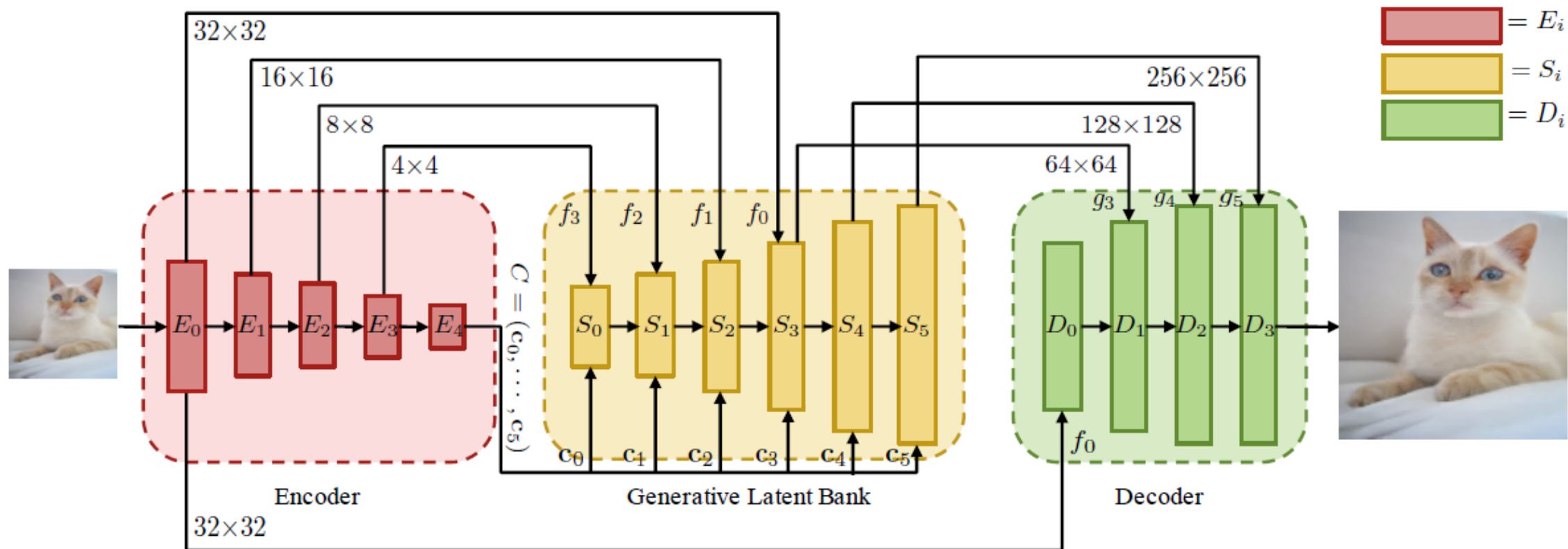
- **Image Super-Resolution**

- 기존 L2 loss를 이용하여 고해상도 이미지를 만드는 것을 perceptual loss를 이용하여 디테일한 고해상도 이미지를 만들 수 있게 개선하였습니다.
- 충실도와 자연스러운 이미지 특성을 학습해야 하므로 여전히 부자연스러운 이미지를 만듭니다.
- GLEAN은 현재 SR모델들을 뛰어넘는 64x 해상도 개선이 가능합니다.

- **GAN Inversion**

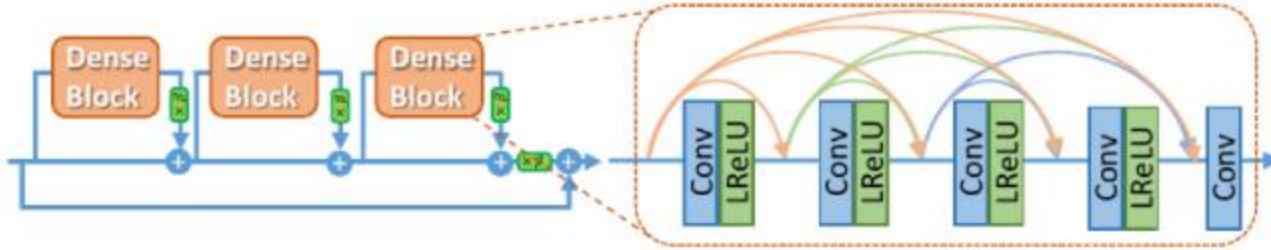
- 저화질 이미지에서 알맞은 latent vector를 만들고 latent vector를 이용하여 고화질 이미지를 만들어 냅니다.
- 낮은 공간 정보를 활용하기 때문에, 정보가 충실하게 유지되지 않습니다.

# GLEAN



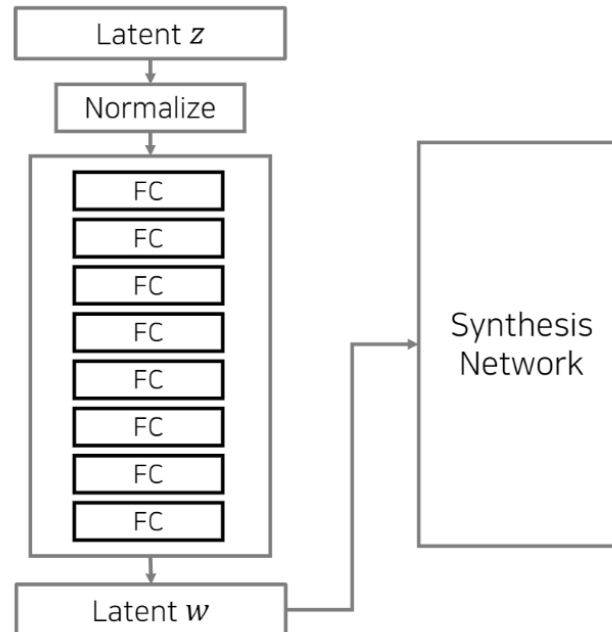
# GLEAN : Encoder

- Residual-in-Residual Dense Block



$$f_i = E_i(f_{i-1}), \quad i \in \{1, \dots, N\},$$

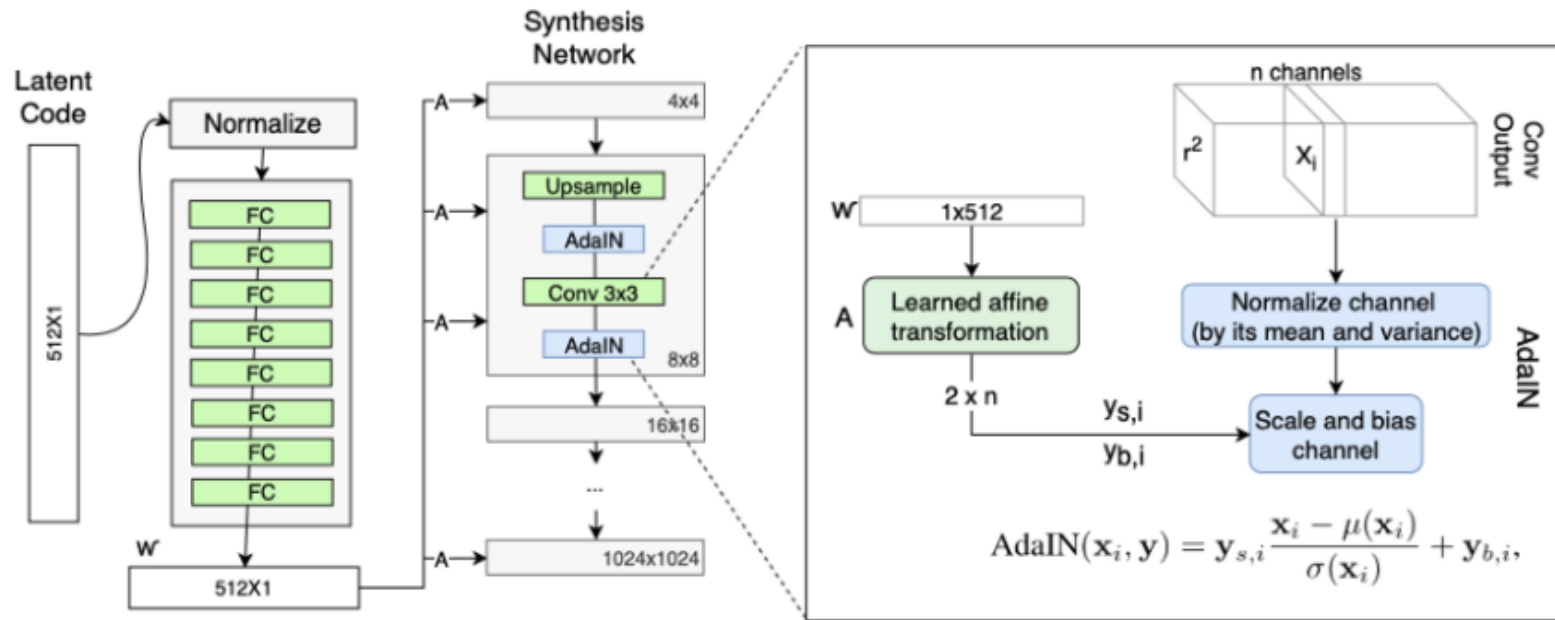
- Mapping Network



$$C = E_{N+1}(f_N),$$

# GLEAN : Generative Latent Bank

- 기존의 Style GAN 모델을 Generative Latent Bank로 사용합니다.

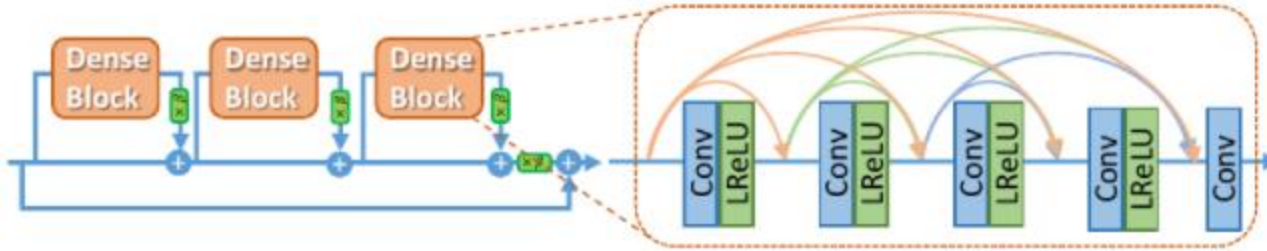


The generator's Adaptive Instance Normalization (AdaIN)

$$g_i = \begin{cases} S_0(c_0, f_N), & \text{if } i = 0, \\ S_i(c_i, g_{i-1}, f_{N-i}), & \text{otherwise,} \end{cases}$$

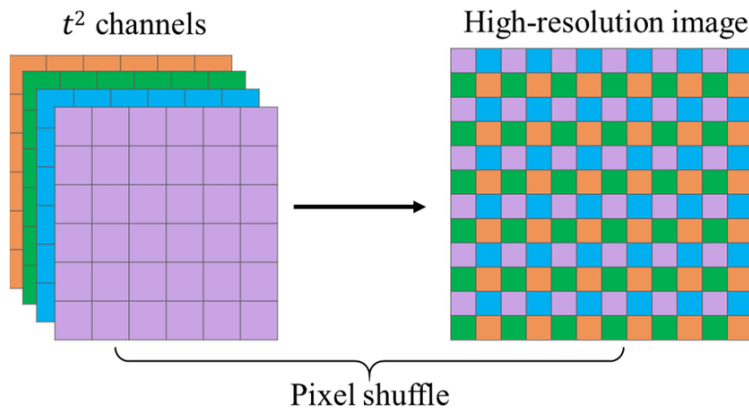
# GLEAN : Decoder

- Residual-in-Residual Dense Block



$$d_i = \begin{cases} D_0(f_0) & \text{if } i = 0, \\ D_i(d_{i-1}, g_{N-2+i}) & \text{otherwise,} \end{cases}$$

- Pixel shuffle layer



- Input:  $(N, C * upscale\_factor^2, H, W)$
- Output:  $(N, C, H * upscale\_factor, W * upscale\_factor)$

# GLEAN : Training

- **Generator Loss function**

1. L2 loss  $\mathcal{L}_{mse} = \frac{1}{N} \|\hat{y} - y\|_2^2,$
2. Perceptual loss  $\mathcal{L}_{percep} = \frac{1}{N} \|f(\hat{y}) - f(y)\|_2^2,$
3. Adversarial loss  $\mathcal{L}_{gen} = \log(1 - D(\hat{y})),$

$$\mathcal{L}_g = \mathcal{L}_{mse} + \alpha_{percep} \cdot \mathcal{L}_{percep} + \alpha_{gen} \cdot \mathcal{L}_{gen}.$$

- **Discriminator Loss function**

$$\mathcal{L}_d = \log(1 - D(\hat{y})) + \log D(y).$$



# GLEAN : Training

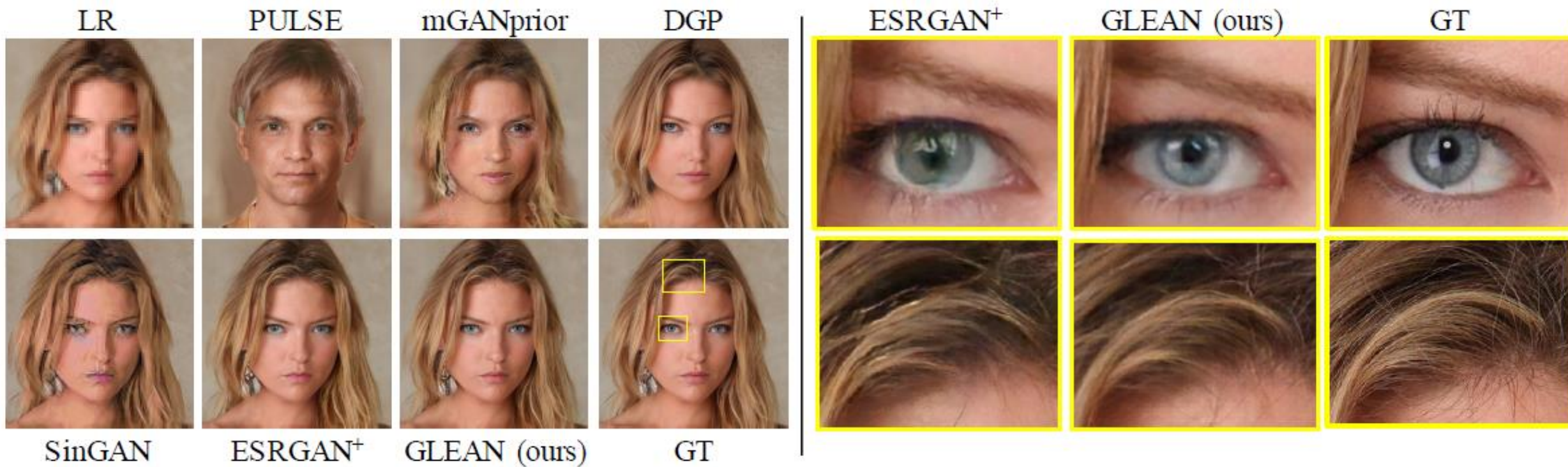
---

- **Parameters**

1. Optimizer : Adam
2. Learning rate :  $10^{-4}$  (cosine annealing)
3. Epochs : 300K
4. Batch size : 8 (human image), 16 (other categories)
5.  $\alpha_{\text{percep}} = \alpha_{\text{gen}} = 10^{-2}$

# Experiment

- Qualitative Comparison



# Experiment

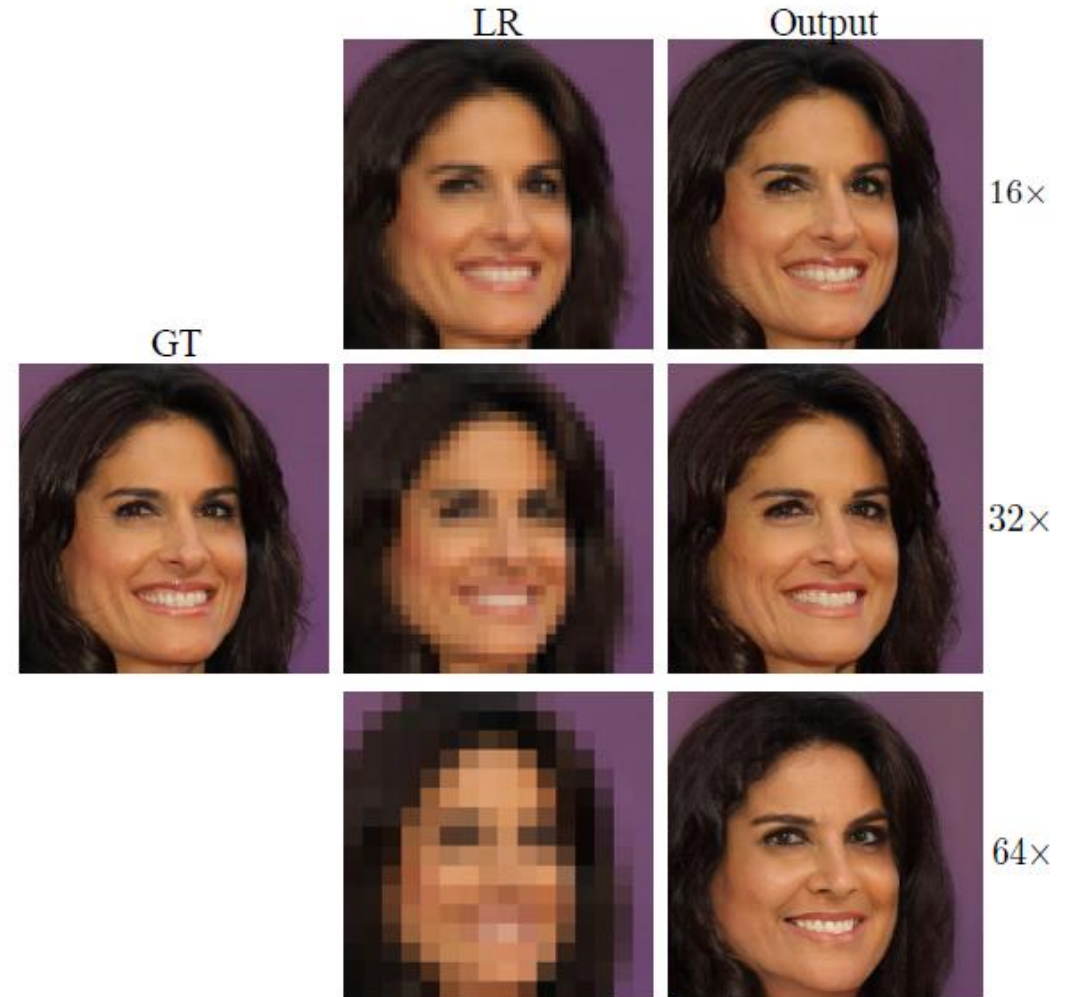
- Qualitative Comparison



# Experiment

## • Qualitative Comparison

- 원본 이미지에 가장 충실하고 자연스러운 이미지를 생성합니다.
- 얼굴이 아닌 다른 카테고리의 이미지에서도 좋은 성능을 보입니다.
- 64x SR 에서도 GT와 비교 했을 때, 지각적으로 비슷한 이미지를 만듭니다.





# Experiment

- Robustness to poses and contents

- 얼굴이 아닌 이미지에서도 좋은 성능을 보여줍니다.
- 정렬된 얼굴이 아닌 경우에도 좋은 성능을 보여줍니다.



# Experiment

## • Quantitative Comparison

- CelebA-HQ 이미지를 이용하여 ArcFace Embedding 공간에서 코사인 유사도가 가장 높게 나옵니다.
- 얼굴이 아닌 다른 이미지에서 PSNR과 LPIPS 를 측정했을 때, 대부분 카테고리에서 가장 우수한 성능을 보입니다.

	PULSE [27]	mGANprior [11]	DGP [28]
Similarity	0.4047	0.5526	0.7341
	SinGAN [30]	ESRGAN <sup>+</sup> [35]	GLEAN
Similarity	0.7718	0.9599	<b>0.9678</b>

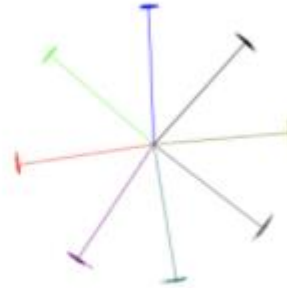
	mGANprior [11]	PULSE [27]	ESRGAN <sup>+</sup> [35]	GLEAN
Face [15]	23.66/0.4661	21.83/0.4600	26.76/0.2787	<b>26.84/0.2681</b>
Cat [43]	17.01/0.5556	19.78/0.5241	19.99/0.3482	<b>20.92/0.3215</b>
Car [20]	14.53/0.7228	16.30/0.6491	19.42/0.3006	<b>19.74/0.2830</b>
Bedroom [41]	16.38/0.5439	12.97/0.7131	<b>19.47/0.3291</b>	19.44/0.3310
Tower [41]	15.96/0.4870	13.62/0.7066	17.86/0.3132	<b>18.41/0.2850</b>

# Experiment

- ArcFace



(a) Softmax



(b) ArcFace

- PSNR & LPIPS

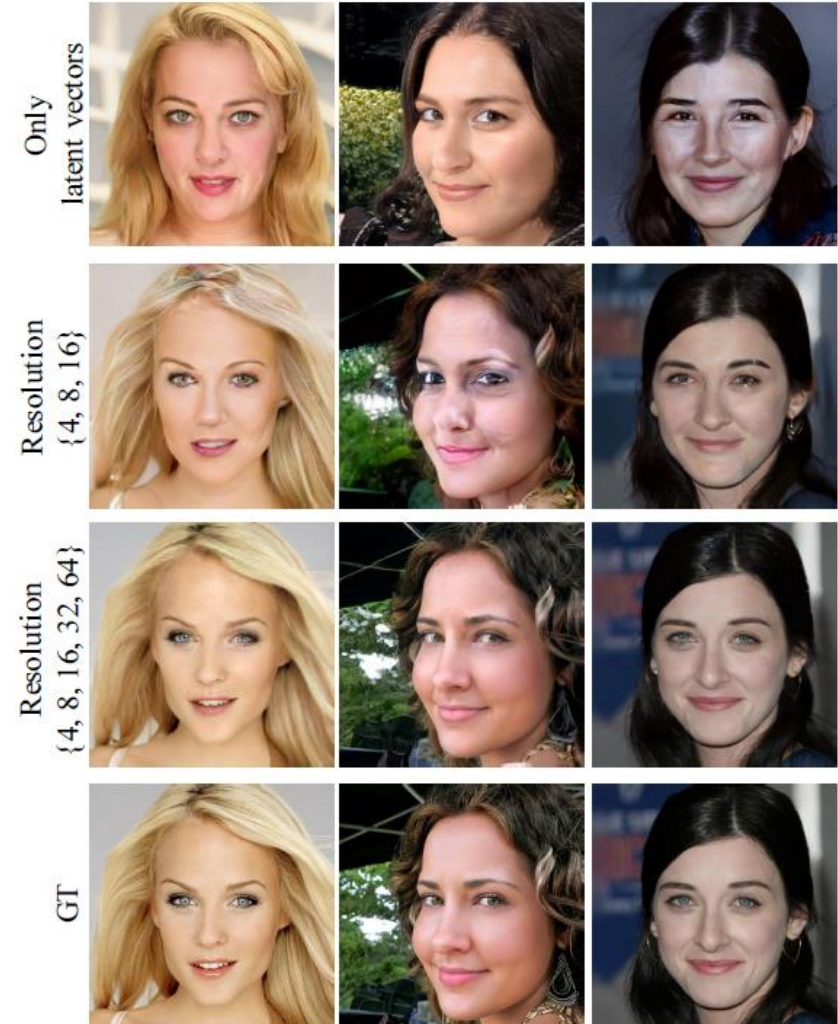
$$\begin{aligned} PSNR &= 10 \cdot \log_{10} \left( \frac{MAX_I^2}{MSE} \right) \\ &= 20 \cdot \log_{10} \left( \frac{MAX_I}{\sqrt{MSE}} \right) \\ &= 20 \cdot \log_{10}(MAX_I) - 10 \cdot \log_{10}(MSE) \end{aligned}$$

$$l_{VGG/i,j}^{SR} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{HR})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{LR}))_{x,y})^2$$

# Ablation Study

- Importance of multi-resolution encoder features

- encoder features 를 사용하지 않았을 때는, 전체적인 특성(머리색, 포즈)만 비슷한 이미지를 만듭니다.
- 점진적으로 encoder features 를 추가하면, 점점 더 디테일한 부분을 생성하고 GT와 가까운 이미지를 만듭니다.
- Latent vector 만으로 만들 수 없는 미세한 부분을 encoder features가 중요한 역할을 합니다.

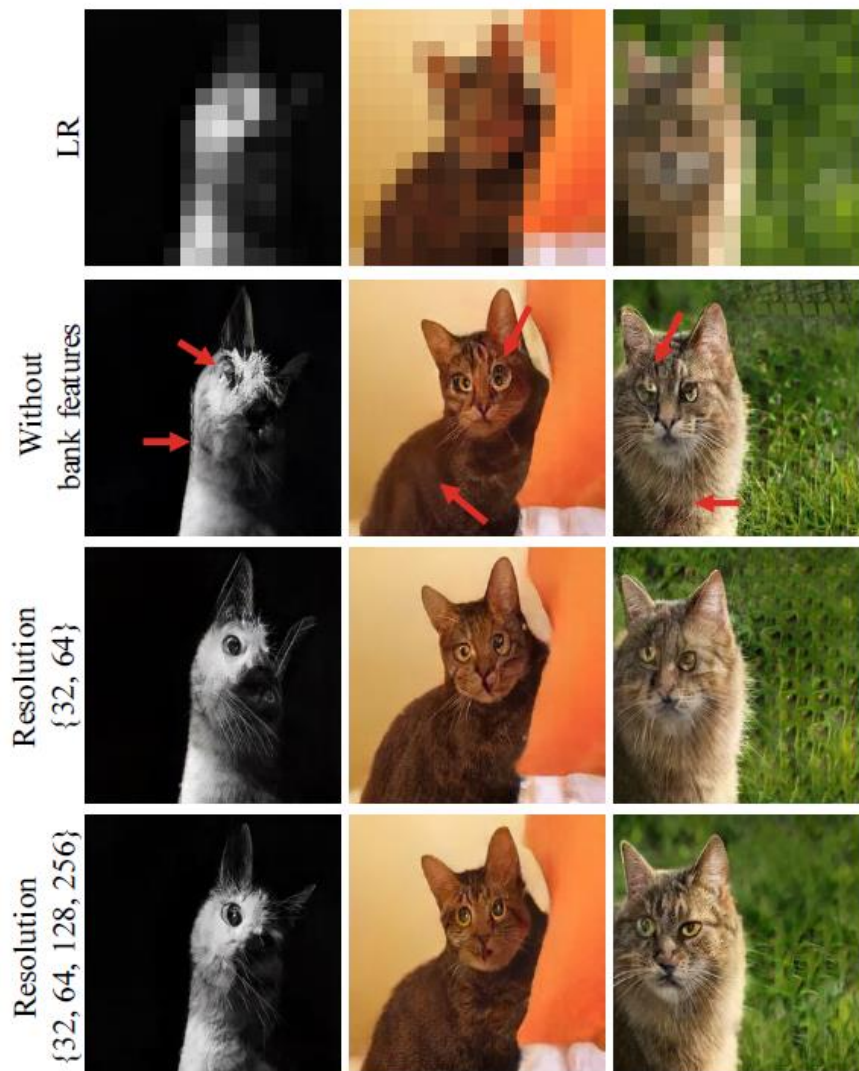




# Ablation Study

- Effects of latent bank features

- Latent bank features를 전달하지 않을 때, 구조 복원과 질감 표현 모두 부족한 이미지를 생성합니다.
- 더 미세한 특성을 decode에 전달할 때, 구조와 질감에서 모두 개선되는 것을 보입니다.



# Ablation Study

- Importance of decoder

- 지각적으로 원본과 비슷한 이미지를 만들지만, 이미지를 확대했을 때 자연스럽지 못한 부분이 보입니다.



# Ablation Study

- **Comparison with reference-based method**

## DFDNet

- DFDNet으로 만든 고화질 이미지는 얼굴 구성요소는 잘 복원하지만, 피부나 머리카락에서는 좋은 결과를 얻지 못합니다.

## SRNTT

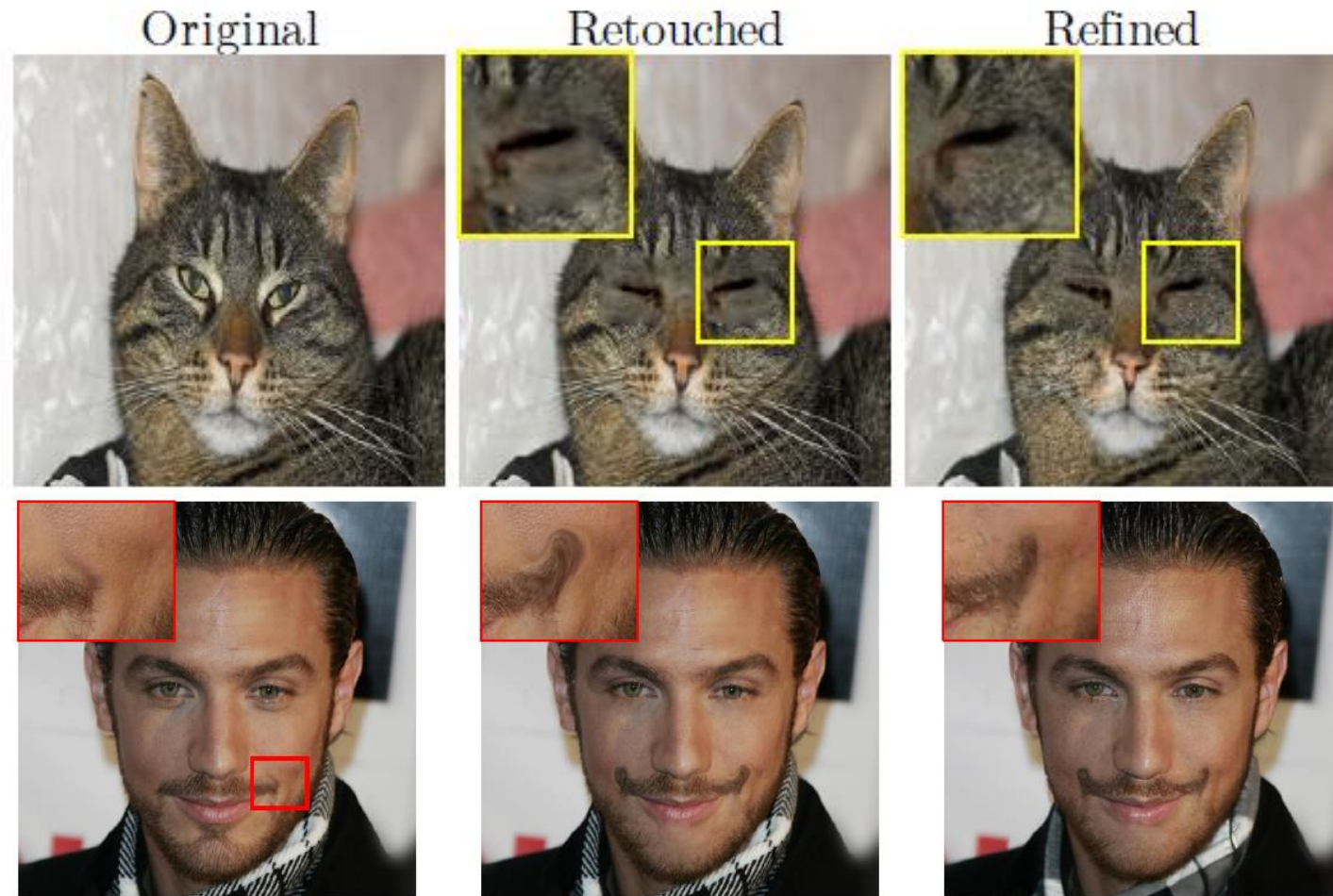
- SRNTT는 전반적으로 흐릿한 이미지를 생성하고 질감을 제대로 표현하지 못합니다.





# Application – Image Retouching

- GLEAN을 Image Retouching 용으로 사용할 수 있습니다.
- 이미지를 원하는 모습으로 대충 retouch를 한 후에 GLEAN 모델을 사용하면 자연스러운 이미지 생성이 가능합니다.
- forward pass를 한번만 거치면 되는 모델이기 때문에 기존의 소프트웨어와 쉽게 통합이 가능합니다.



# Conclusion

---

- 기존에 학습된 GAN을 활용하여 64x SR 모델을 제시했습니다.
- latent bank를 사용하는 효과적인 방법 덕분에 한번의 forward pass로 고화질 이미지 생성이 가능합니다.
- GLEAN은 이미지 노이즈 제거, 이미지 채색 등 여러 작업으로 확장이 가능한 모델입니다.

---

**Thank you**

**Q & A**

---