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

2021 VILlab Summer Seminar

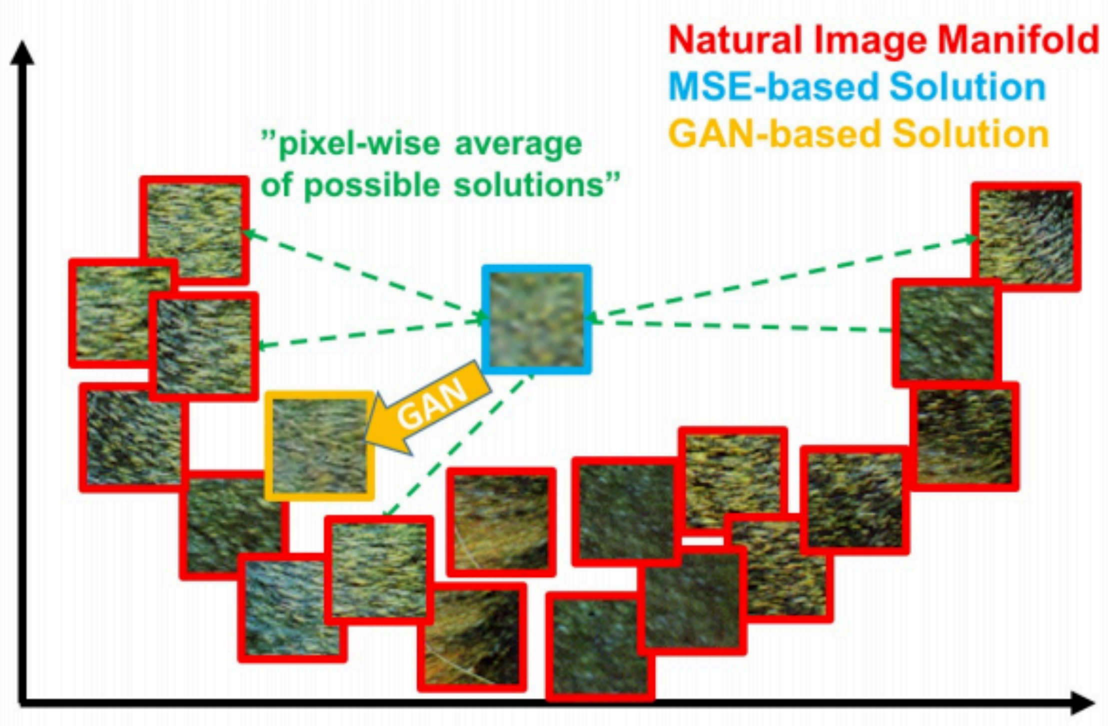
Seobin Park

Content

1. Preliminaries
    1. GAN Recap
    2. Simple GAN-based SR
    3. GAN Inversions for SR
2. GLEAN
    1. Method
    2. Network Architecture
    3. Experimental Results
3. Potential Applications

Preliminaries: GAN Recap

- We want to sample a natural image from p_{data}.
- But this sampling process is intractable. So we try to sample natural images from a tractable random variable z and a neural network G.



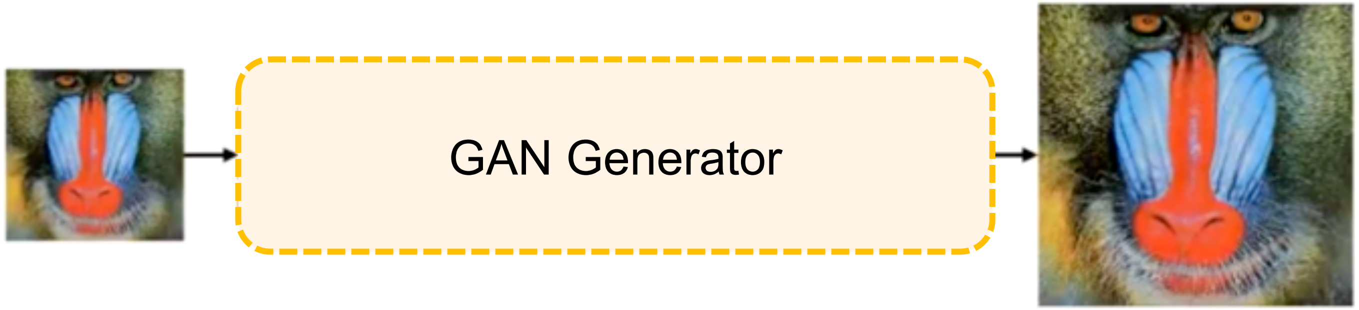
$z \xrightarrow[\mathbf{G}]{}$ Natural image

Preliminaries: GAN Recap

- Define **probability distribution** $\boldsymbol{p_G}$ as the distribution of the samples obtained from G(z). (G is a neural network and z is a sample from a tractable distribution like Gaussian.)

- GAN's minimax game has a **global optimum for** $\boldsymbol{p_G = p_{data}}$.

Preliminaries: Simple GAN-based SR



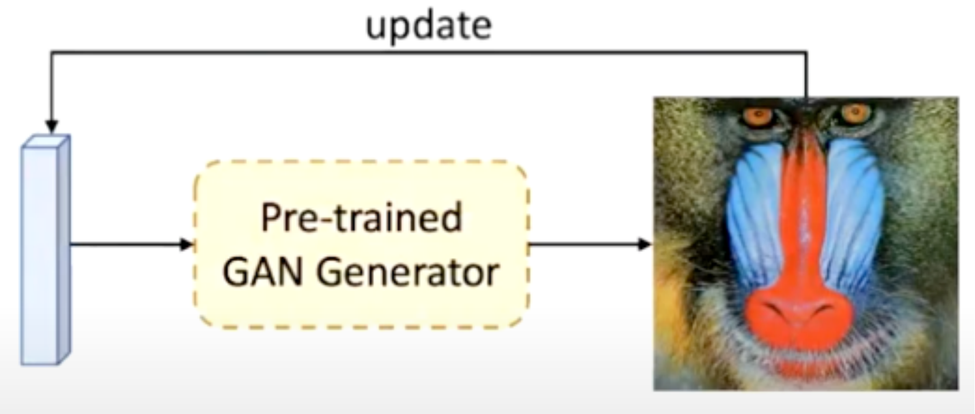
$$\begin{cases} l_{MSE}^{SR} = \dfrac{1}{r^2WH}\displaystyle\sum_{x=1}^{rW}\sum_{y=1}^{rH}(I_{x,y}^{HR} - G_{\theta_G}(I^{LR})_{x,y})^2 \\[2em] l_{Gen}^{SR} = \displaystyle\sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR})) \end{cases}$$

$\Rightarrow$ **Fidelity loss**

$\Rightarrow$ (Must not deviate too much from original HR image)

$\Rightarrow$ **Limits the capability of approximating the natural image manifold on challenging scale factors (x8 ~ x64)**

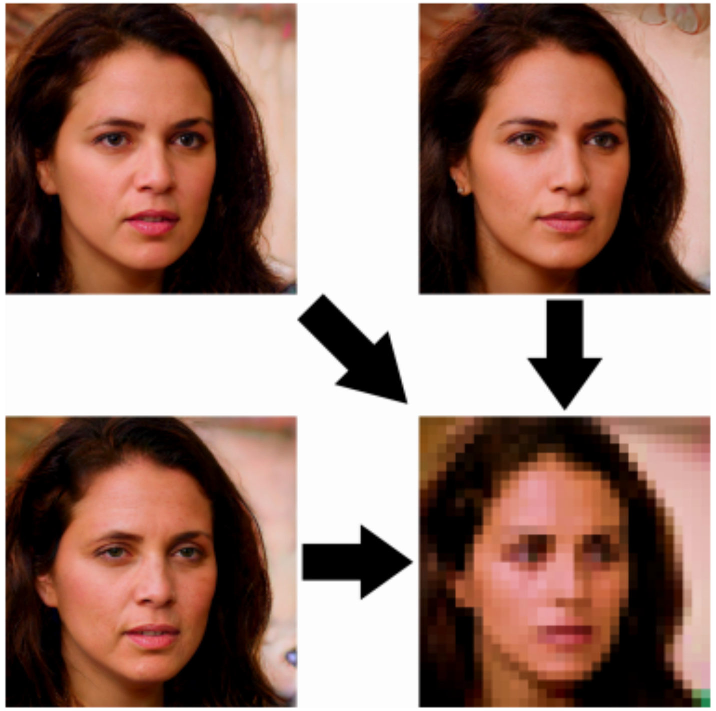
Preliminaries: GAN Inversions for SR



$$z^* = \mathrm{argmin}_{z \in \mathcal{Z}} \underline{\mathcal{L}\left(G(z), x\right)}$$

Optimizie a task-specific loss
to impose a **strong prior**

Preliminaries: PULSE



$$z^* = \mathrm{argmin}_{z \in \mathcal{Z}} \underline{\mathcal{L}(G(z), x)}$$

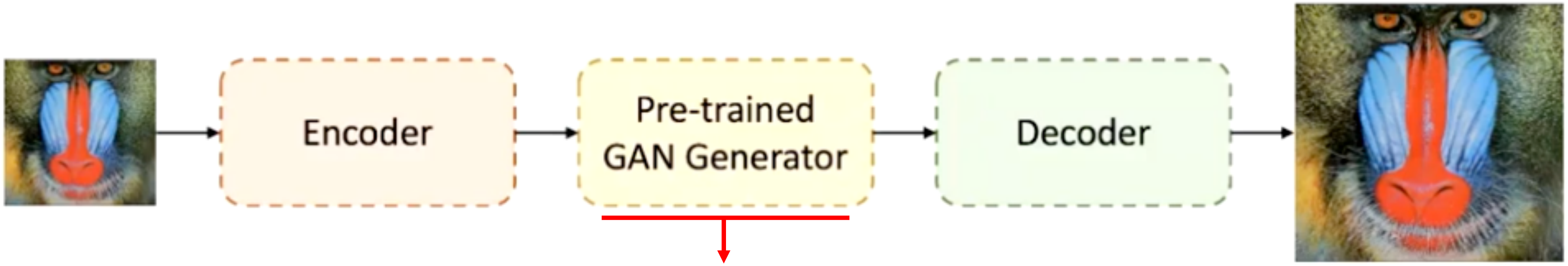
$$\underline{L_{DS}(I_{SR}, I_{LR})} := \|DS(I_{SR}) - I_{LR}\|_p^p$$

Problems:

$\Rightarrow$ **Insufficient** to guide restoration process

$\Rightarrow$ **Time consuming** at test time

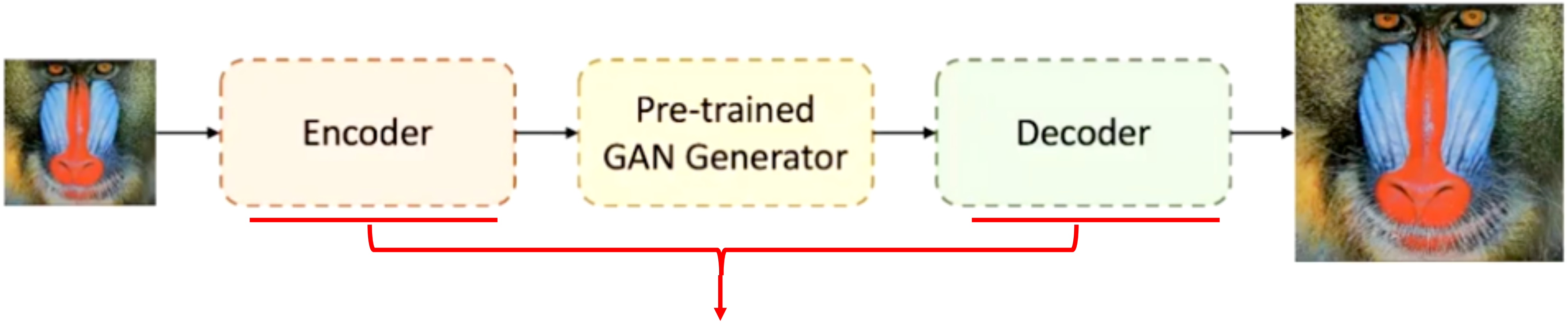
GLEAN: Method



- Use a pre-trained GAN generator to impose the prior
- This generator is trained **without any fidelity loss**
- This generator works as a " **GAN-based dictionary** " (or a " latent bank ")

$$l^{SR}_{MSE} = \frac{1}{r^2WH}\sum_{x=1}^{rW}\sum_{y=1}^{rH}(I^{HR}_{x,y} - G_{\theta_G}(I^{LR})_{x,y})^2 \longrightarrow \text{X}$$

GLEAN: Method



- Pre-trained GAN generator is fixed in the training process.
- Use the same dataset that is used to train pre-trained GAN Generator.
- Conventional GAN-based SR loss is used for training.

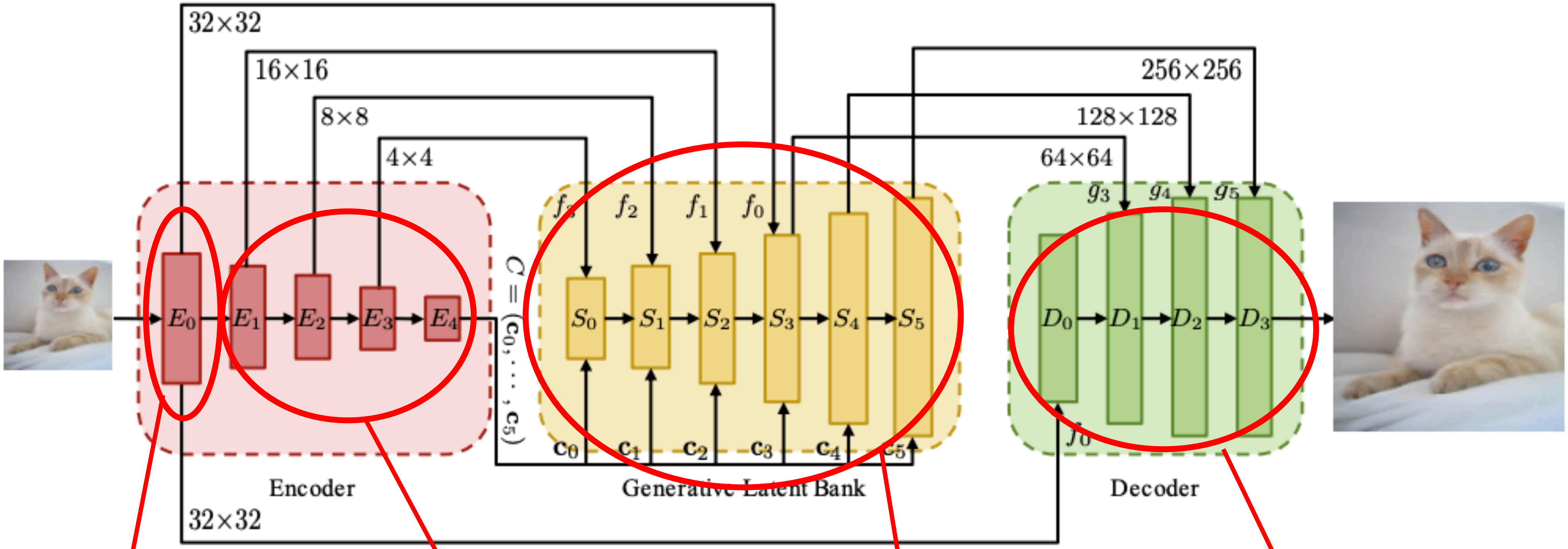
$$\mathcal{L}_g = \mathcal{L}_{mse} + \alpha_{percep} \cdot \mathcal{L}_{percep} + \alpha_{gen} \cdot \mathcal{L}_{gen}$$

GLEAN: Pros

1. No need for time-consuming optimizations at runtime. Instead, only one forward pass is needed.

2. Pre-trained GANs is employed as a latent bank (dictionary) to impose stronger prior.

GLEAN: Network Architecture



ESRGAN architecture (RRDB-Net)

Stride2 conv + stride1 conv

StyleGAN architecture

3x3 convs

GLEAN: Experimental Results

(a) Low-Resolution (b) ESRGAN (c) PULSE (d) GLEAN (ours) (e) Ground-truth

Figure 1: **Example of large-factor super-resolution (16×).**

GLEAN: Experimental Results

Table 1: **Cosine similarity of ArcFace features [5] for 16× SR.** GLEAN achieves a higher similarity than baselines. **Bolded** texts represent the best performance.

	PULSE [26]	mGANprior [11]	DGP [27]
Similarity	0.4047	0.5526	0.7341

	SinGAN [29]	ESRGAN$^+$ [33]	**GLEAN**
Similarity	0.7718	0.9599	**0.9678**

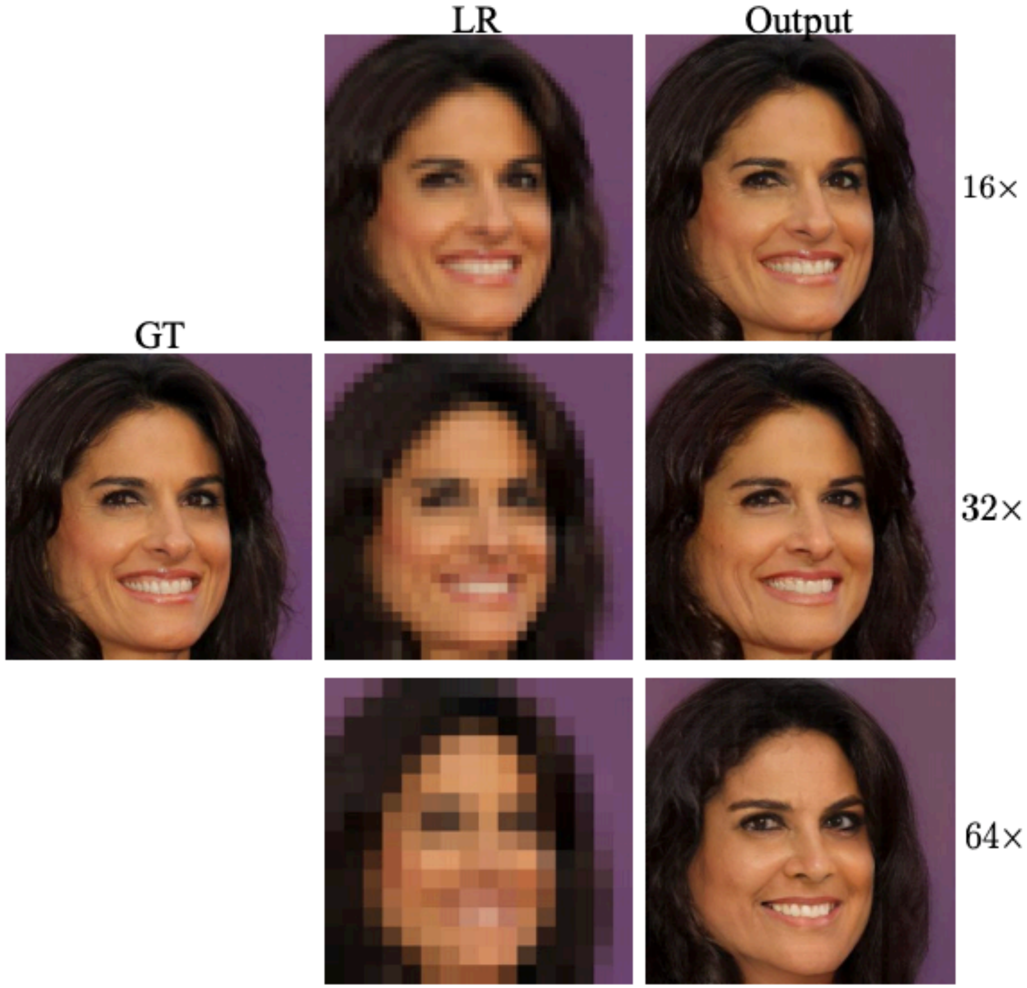
GLEAN: Experimental Results

Table 2: **Quantitative (PSNR/LPIPS) comparison on 16× SR.** GLEAN outperforms other methods in most categories. ESRGAN$^+$ denotes a larger version of ESRGAN [33] having similar FLOPs to GLEAN. **Bolded** texts represent the best performance.

	mGANprior [11]	PULSE [26]	ESRGAN$^+$ [33]	**GLEAN**
Face [16]	23.66/0.4661	21.83/0.4600	26.76/0.2787	**26.84/0.2681**
Cat [41]	17.01/0.5556	19.78/0.5241	19.99/0.3482	**20.92/0.3215**
Car [20]	14.53/0.7228	16.30/0.6491	19.42/0.3006	**19.74/0.2830**
Bedroom [39]	16.38/0.5439	12.97/0.7131	**19.47/0.3291**	19.44/0.3310
Tower [39]	15.96/0.4870	13.62/0.7066	17.86/0.3132	**18.41/0.2850**

=> High fidelity + Photo-realistic!

GLEAN: Experimental Results



GLEAN: Experimental Results

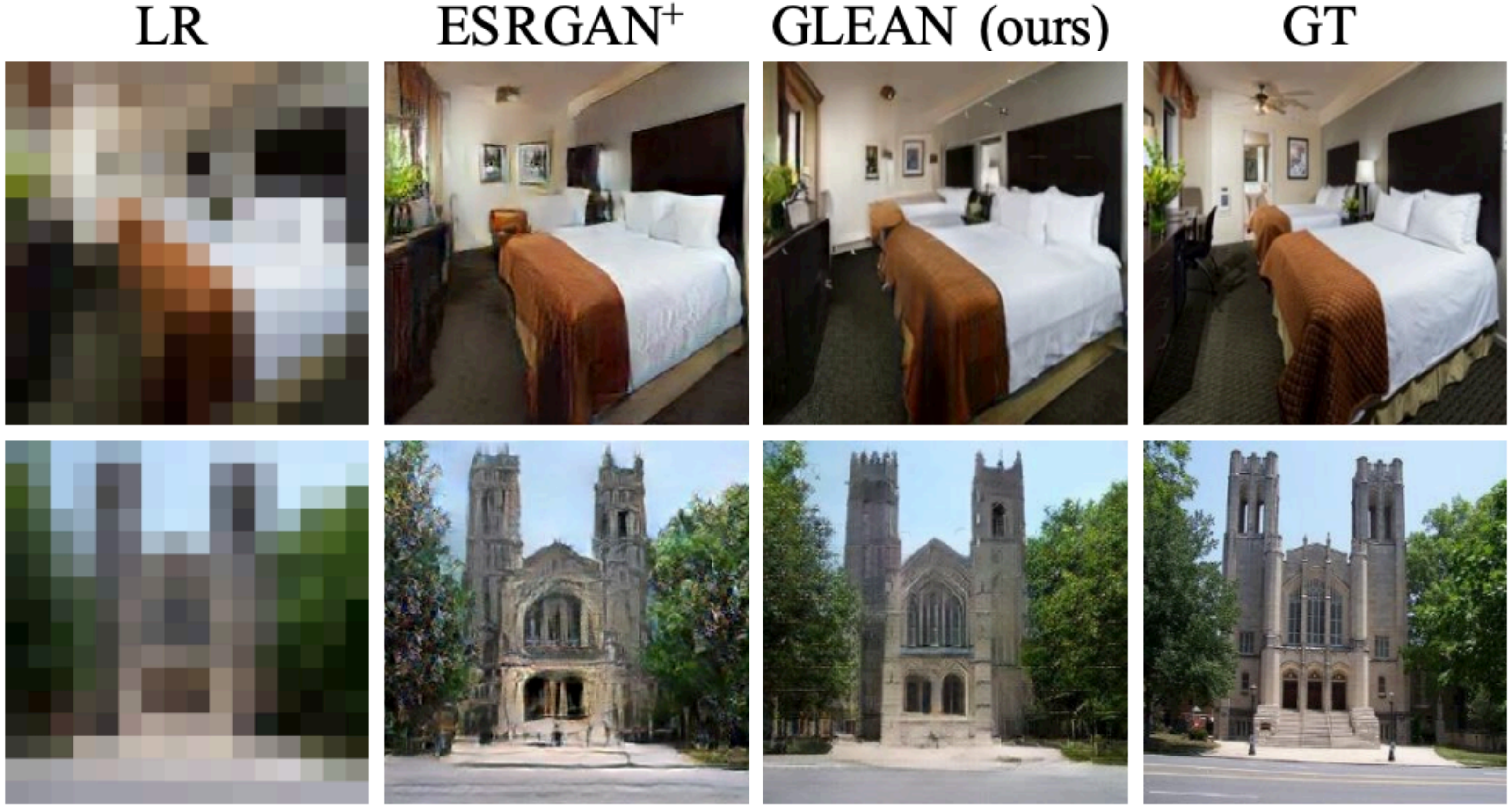
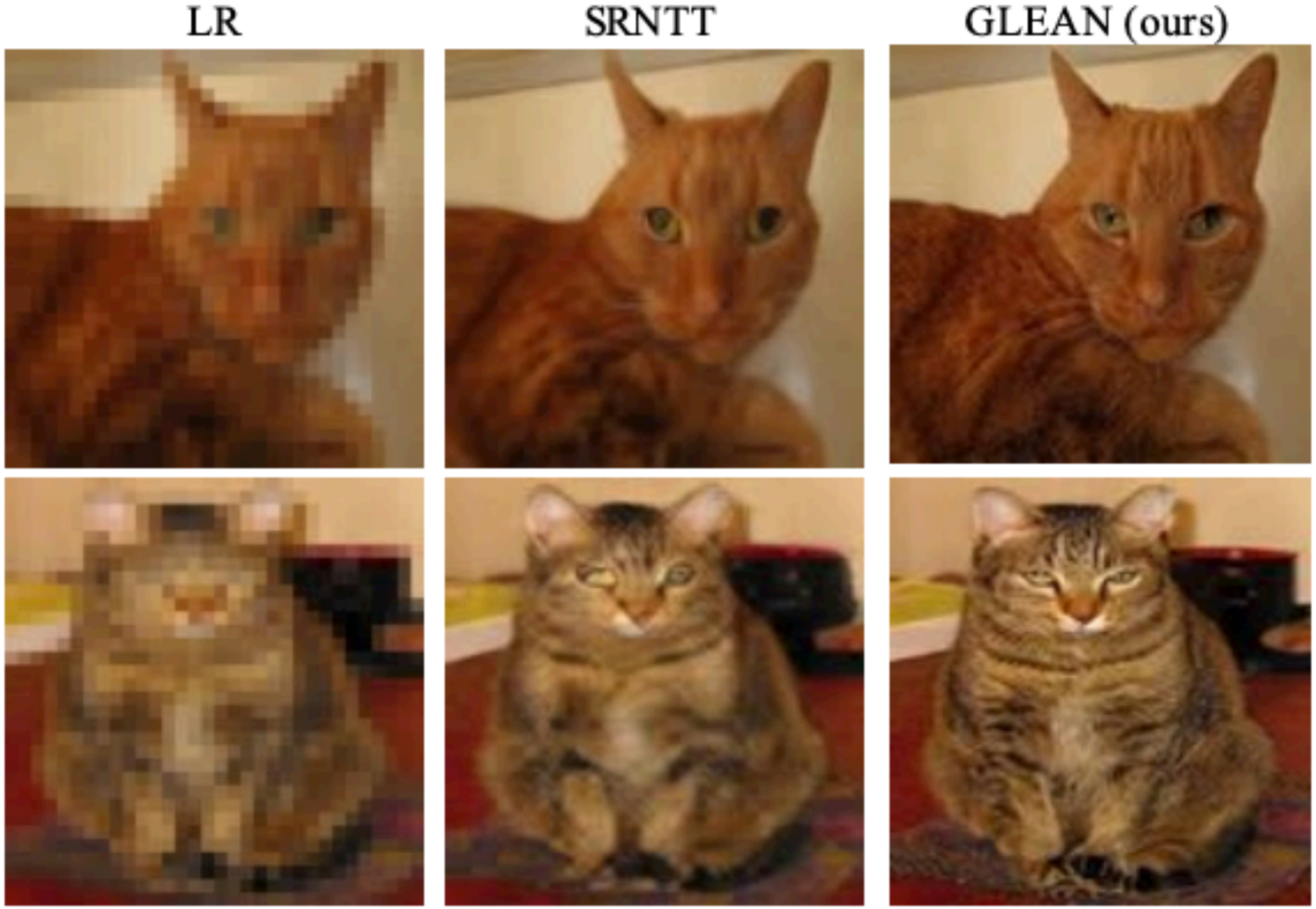


Figure 4: **Results of 16× SR on other categories.**

GLEAN: Experimental Results

(a) Comparison with DFDNet [23]

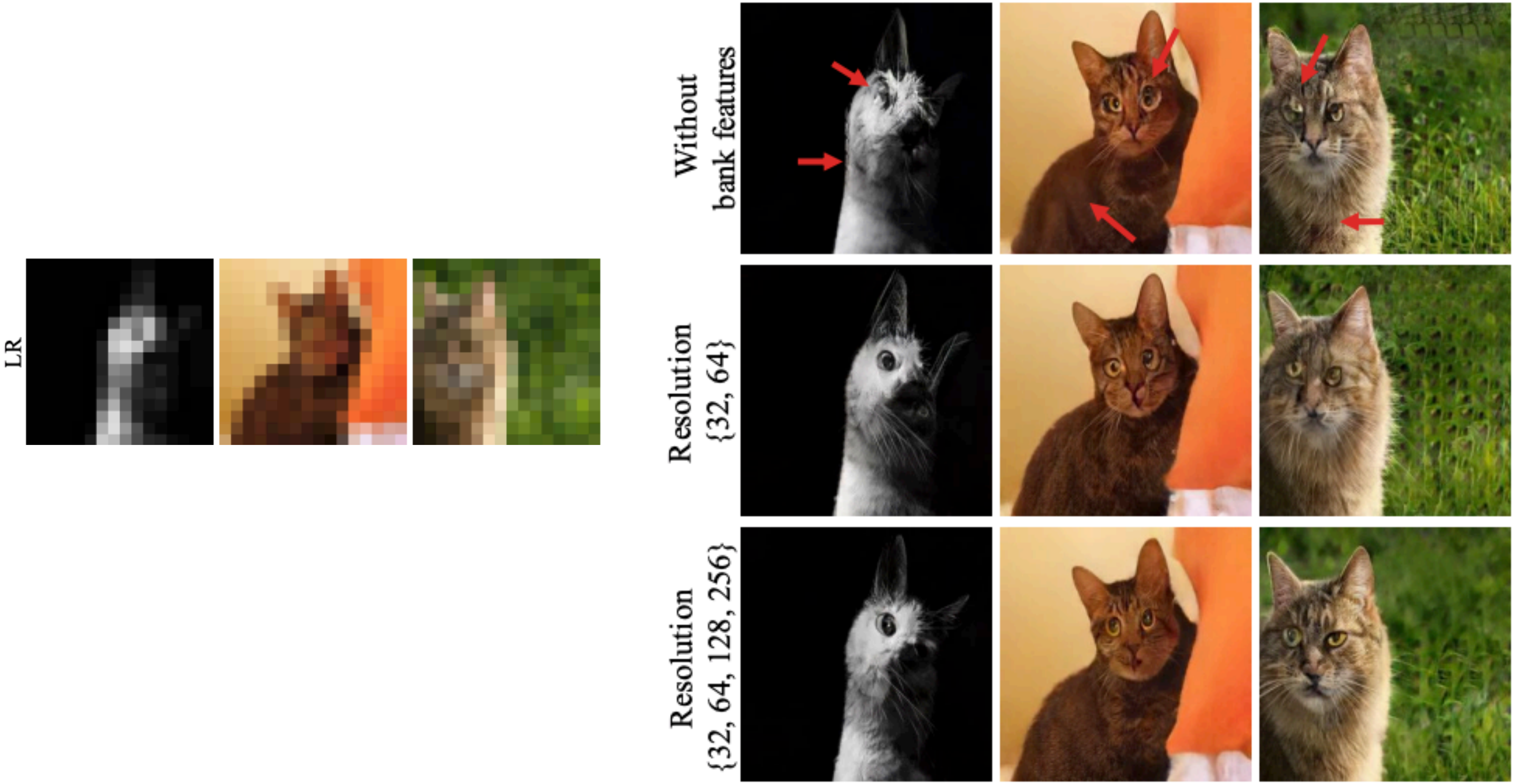
GLEAN: Experimental Results



(b) Comparison with SRNTT [45]

GLEAN: Experimental Results

GLEAN: Experimental Results



Potential Applications

1. Other restoration tasks such as colorization and denoising.

2. Meta-Learned latent bank

References

- Chan, Kelvin CK, et al. "Glean: Generative latent bank for large-factor image super-resolution." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
- Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems* 27 (2014).
- Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
- Karras, Tero, Samuli Laine, and Timo Aila. "A style-based generator architecture for generative adversarial networks." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.
- Menon, Sachit, et al. "Pulse: Self-supervised photo upsampling via latent space exploration of generative models." *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*. 2020.
- Wang, Xintao, et al. "Esrgan: Enhanced super-resolution generative adversarial networks." *Proceedings of the European conference on computer vision (ECCV) workshops*. 2018.
- https://www.youtube.com/watch?v=73EqkLim41U