Image Super-Resolution via Iterative Refinement

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, Mohammad Norouzi
{sahariac,jonathanho,williamchan,salimans,davidfleet,mnorouzi}@google.com
Google Research, Brain Team

Abstract

We present SR3, an approach to image Super-Resolution via Repeated Refinement. SR3 adapts denoising diffusion probabilistic models [17, 48] to conditional image generation and performs super-resolution through a stochastic iterative denoising process. Output generation starts with pure Gaussian noise and iteratively refines the noisy output using a U-Net model trained on denoising at various noise levels. SR3 exhibits strong performance on super-resolution tasks at different magnification factors, on faces and natural images. We conduct human evaluation on a standard 8× face super-resolution task on CelebA-HQ, comparing with SOTA GAN methods. SR3 achieves a fool rate close to 50%, suggesting photo-realistic outputs, while GANs do not exceed a fool rate of 34%. We further show the effectiveness of SR3 in cascaded image generation, where generative models are chained with super-resolution models, yielding a competitive FID score of 11.3 on ImageNet.

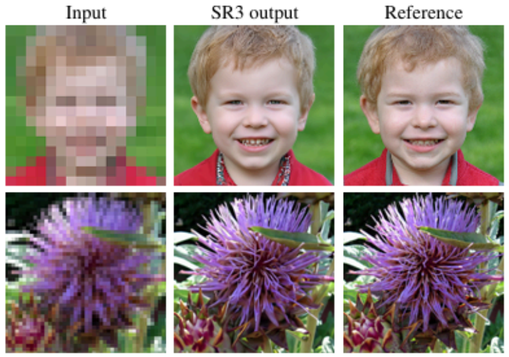


Figure 1: Two representative SR3 outputs: (top) 8× face super-resolution at 16×16→128×128 pixels (bottom) 4× natural image super-resolution at 64×64→256×256 pixels.

Chitwan Saharia† ,Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, Mohammad Norouzi
Google Brain team

Paper Information

· Published June 2021 (IEEE Transactions on Pattern Analysis and Machine Intelligence)

· 76 References, 511 Citations

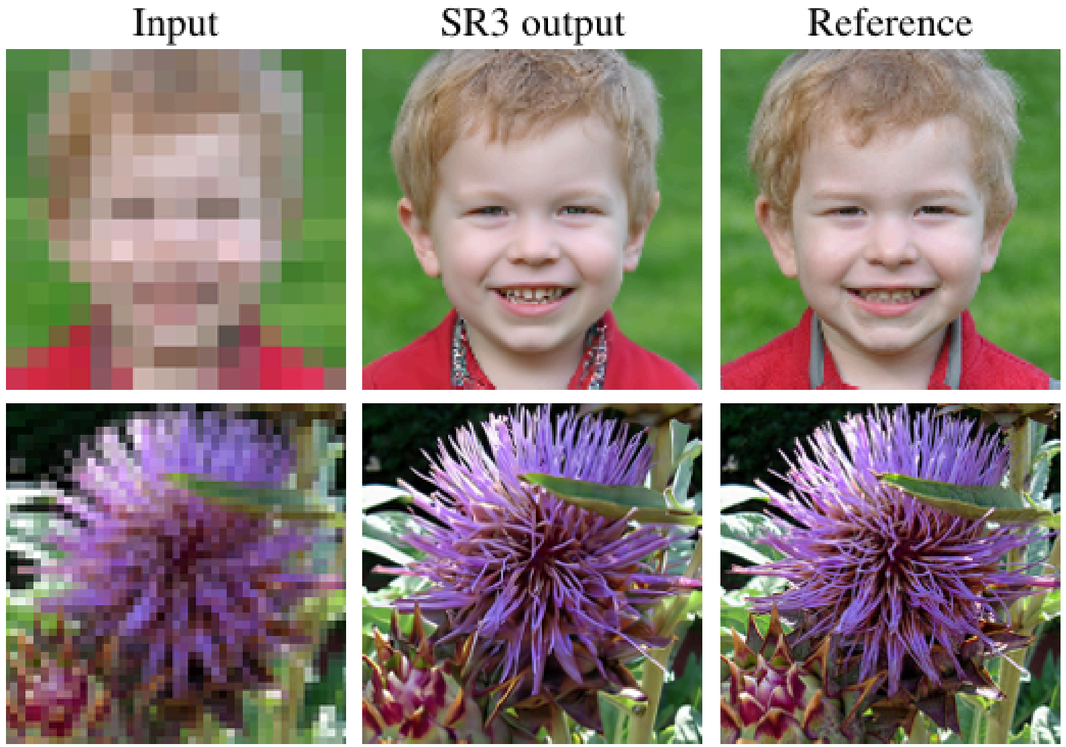
· Paper:
https://arxiv.org/pdf/2104.07636.pdf

it is a foundational paper for application of diffusion models for image super resolution

SR3 adapts denoising diffusion probabilistic models to conditional image gener ation and performs super-resolution through a stochastic iterative denoising process.

related work

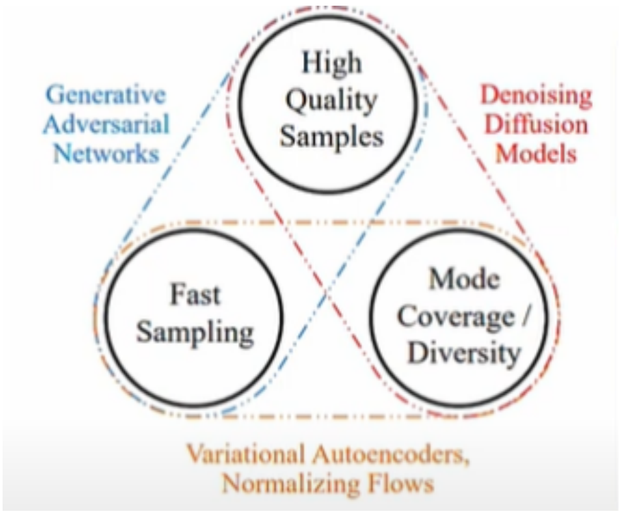
Single-image super-resolution is the process of generat ing a high-resolution image that is consistent with an in put low-resolution image.



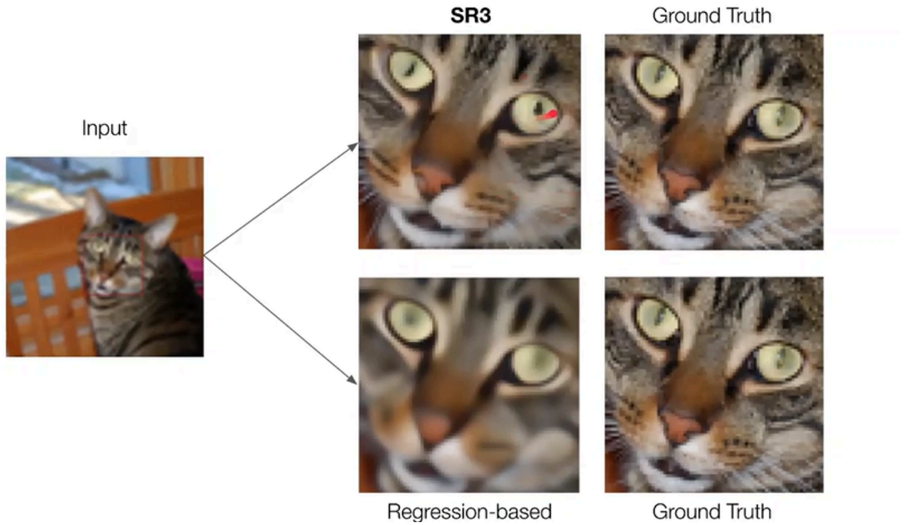
RELATED WORK

while simple regression-based methods with feedfor ward convolutional nets may work for super-resolution at low magnification ratios, they often lack the high-fidelity details needed for high magnification ratios.

Autoregressive models , variational autoencoders (VAEs) Normalizing Flows (NFs) and GANs have shown convincing image generation results and have been applied to conditional tasks such as image super-resolution

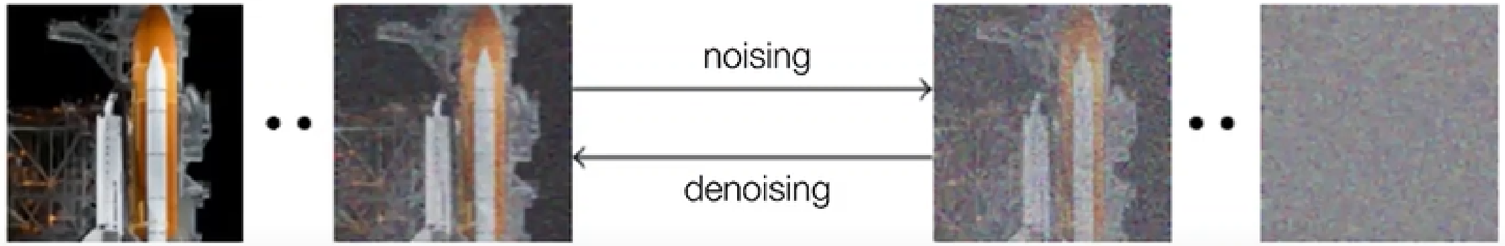


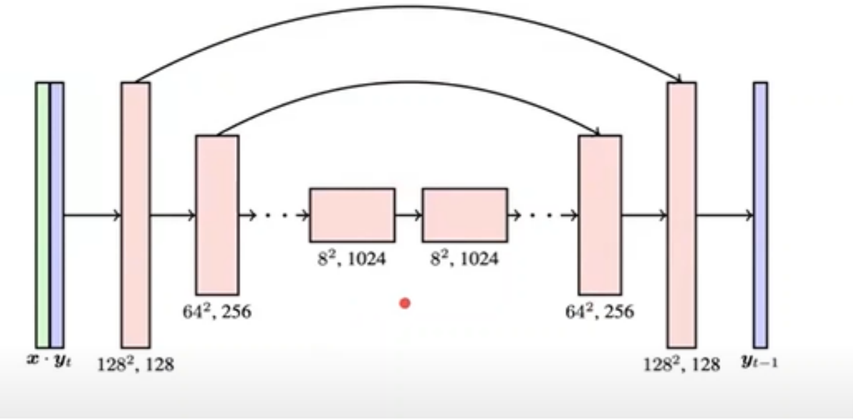
- autoregressive models are prohibitively expensive for high resolution image generation
- NFs and VAEs often yield sub-optimal sample quality
- GANs require carefully de signed regularization and optimization tricks to tame optimization instability and mode collapse

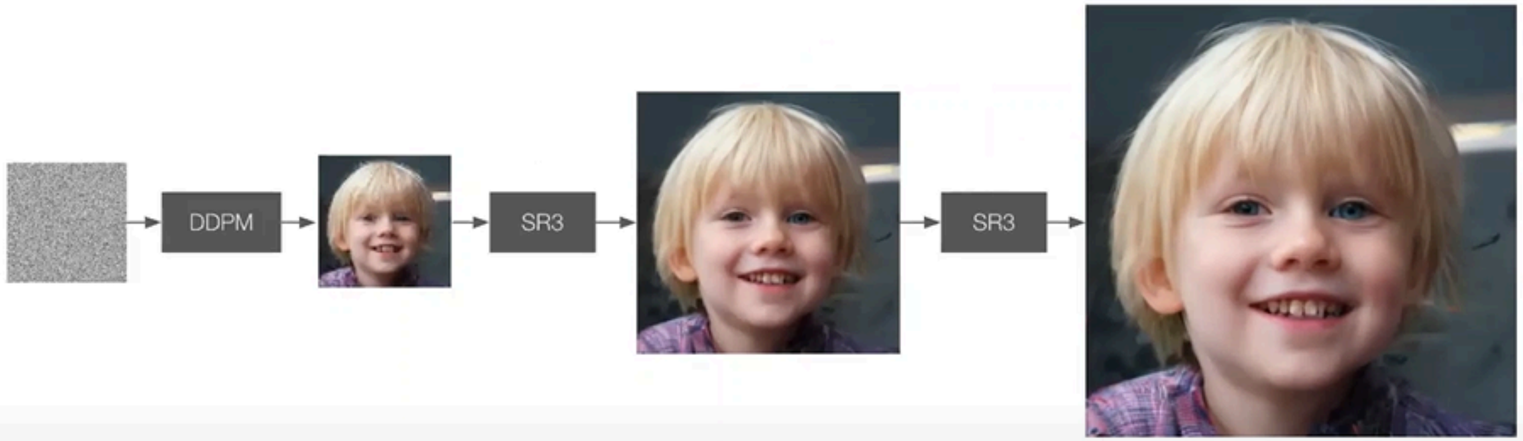


MAIN CONTRIBUTIONS OF PAPER AND FRAMEWORK

- Modified U-Net residual blocks
- Iterative refinement steps
- Efficient noise sheduling
- Denoising score matching
- Inspired unconditional image synthesis
- demonstrateunconditional and class-conditional generation by cascading to generate high fidelity images

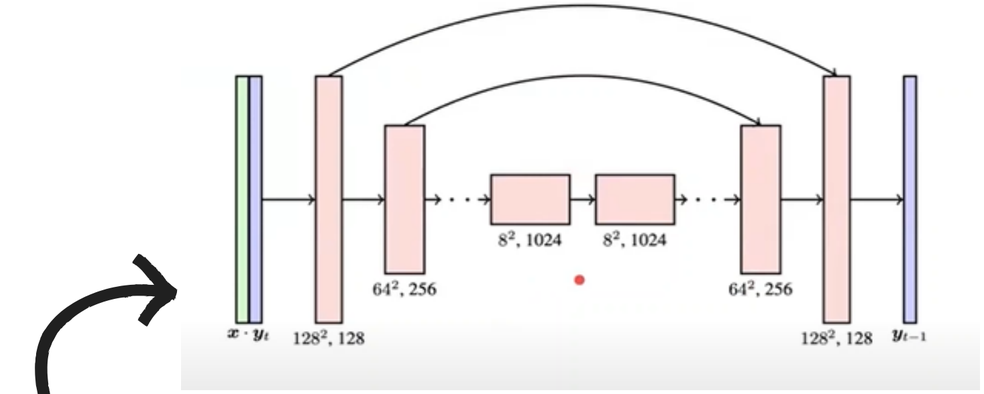


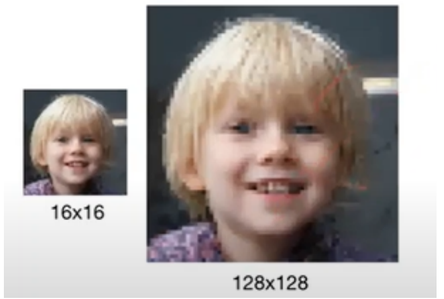


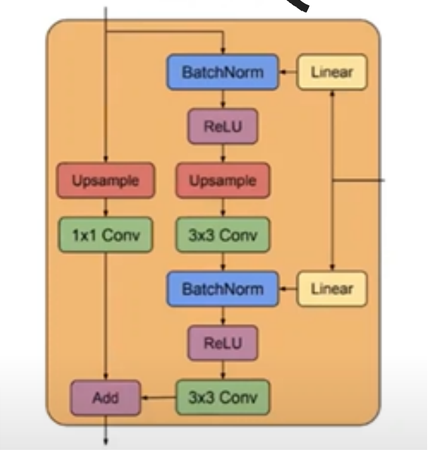


Architecture

- Low-resolution input is upsampled to target resolution using bicubic interpolation.
- Using simple concatenation, concatenate x with target resolution noisy image.
- adopt G blocks in DDPM residual blocks
    - Multi-scale features
    - Spatial Attention
    - Conditioning on additional inputs

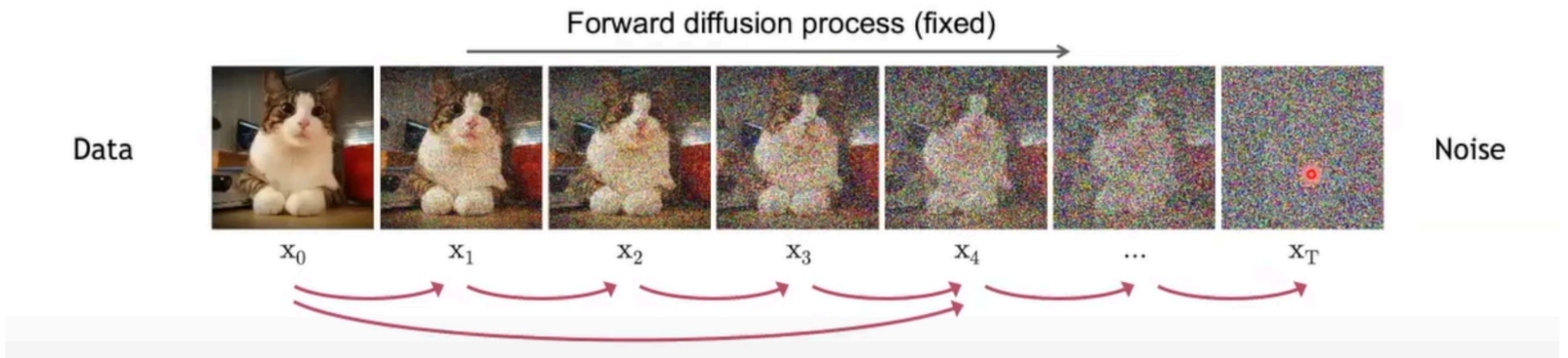






DIFFUSION PROCESS

FORWARD DIFFUSION PROCESS



$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}),$$

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}).$$

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$$

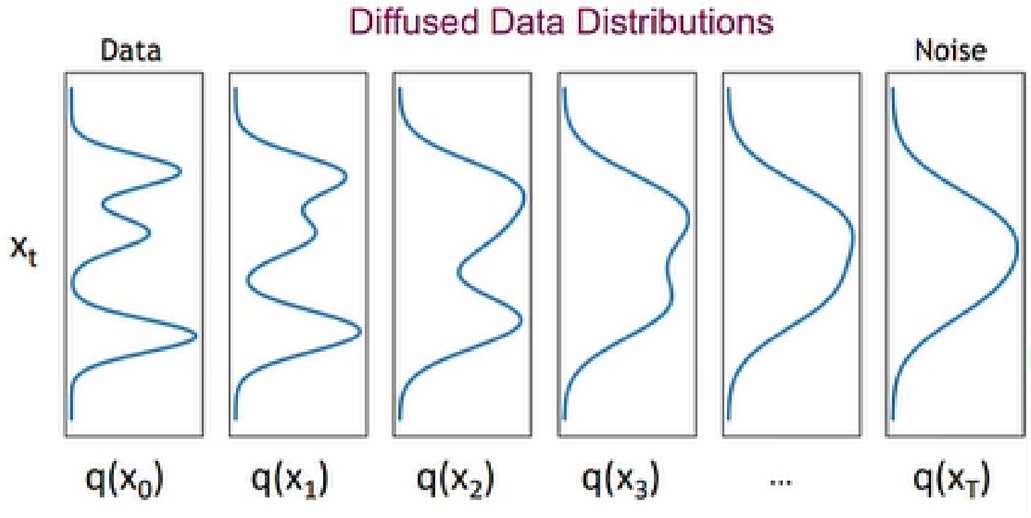
$$\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$$

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon,$$

DIFFUSION PROCESS

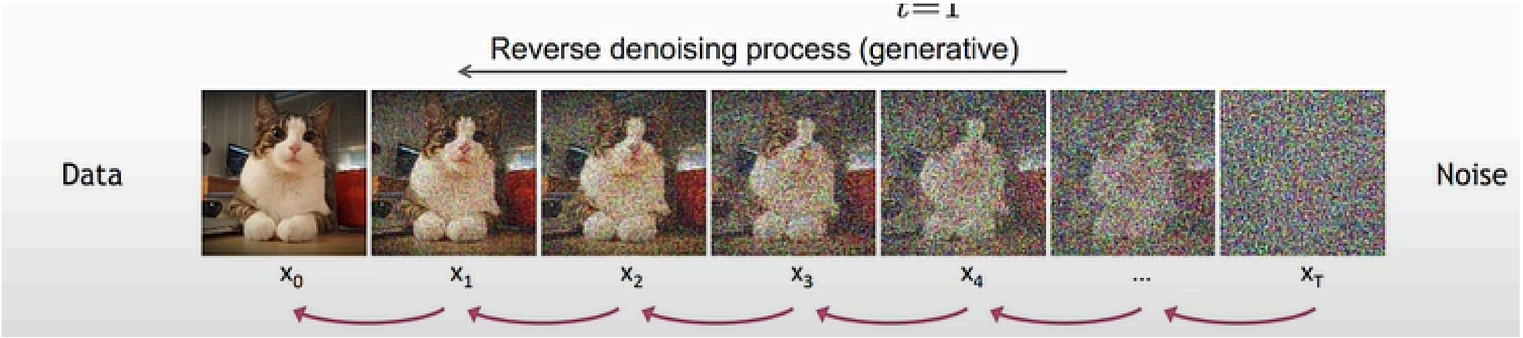
FORWARD DIFFUSION PROCESS

Timestep	Scale Factor	Noise	Noisy Data
0	-	-	5
1	$\sqrt{1-0.2}$	ϵ_1	$5\sqrt{1-0.2}+\sqrt{0.2}\epsilon_1$
2	$\sqrt{1-0.2}$	ϵ_2	$5(\sqrt{1-0.2})^2+\sqrt{0.2}(\epsilon_1+\epsilon_2)$
...	...	...	...
T	$\sqrt{1-0.2}$	ϵ_T	$5(\sqrt{1-0.2})^T+\sqrt{0.2}\sum_{t=1}^{T}\epsilon_t$



DIFFUSION PROCESS

BACKWARD DIFFUSION PROCESS



$$p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$$

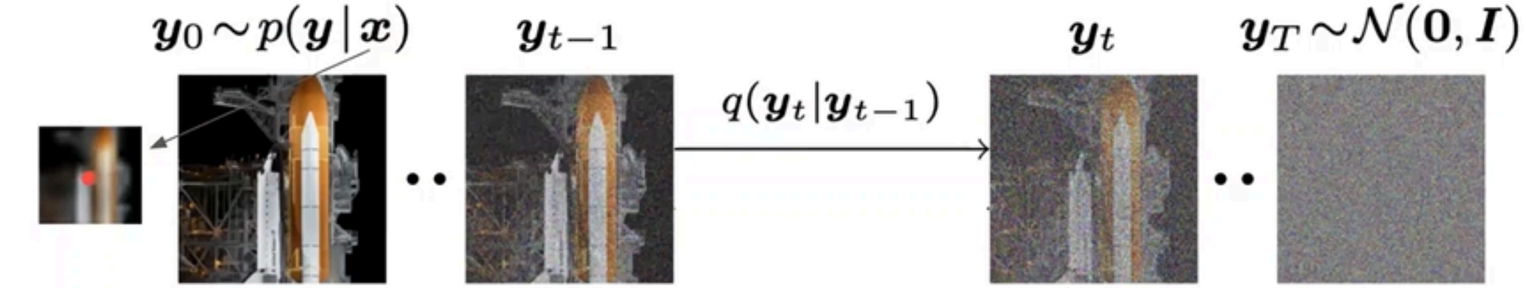
$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} x_0 + \sqrt{1-\bar{\alpha}_{t-1}}\epsilon'$$

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{1-\beta_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right)$$

Timestep	Scale Factor	Noise	Noisy Data
0	-	-	5
1	$\sqrt{1-0.2}$	ϵ_1	$5\sqrt{1-0.2} + \sqrt{0.2}\epsilon_1$
2	$\sqrt{1-0.2}$	ϵ_2	$5(\sqrt{1-0.2})^2 + \sqrt{0.2}(\epsilon_1 + \epsilon_2)$
...	...	...	...
T	$\sqrt{1-0.2}$	ϵ_T	$5(\sqrt{1-0.2})^T + \sqrt{0.2}\sum_{t=1}^{T} \epsilon_t$

Conditional Denoising Diffusion Model



$$q(\boldsymbol{y}_{1:T} \mid \boldsymbol{y}_0) = \prod_{t=1}^{T} q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1})$$

$$q(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}) = \mathcal{N}(\boldsymbol{y}_t \mid \sqrt{\alpha_t}\,\boldsymbol{y}_{t-1}, (1-\alpha_t)\boldsymbol{I})$$

$$q(\boldsymbol{y}_t \mid \boldsymbol{y}_0) = \mathcal{N}(\boldsymbol{y}_t \mid \sqrt{\gamma_t}\,\boldsymbol{y}_0, (1-\gamma_t)\boldsymbol{I})$$

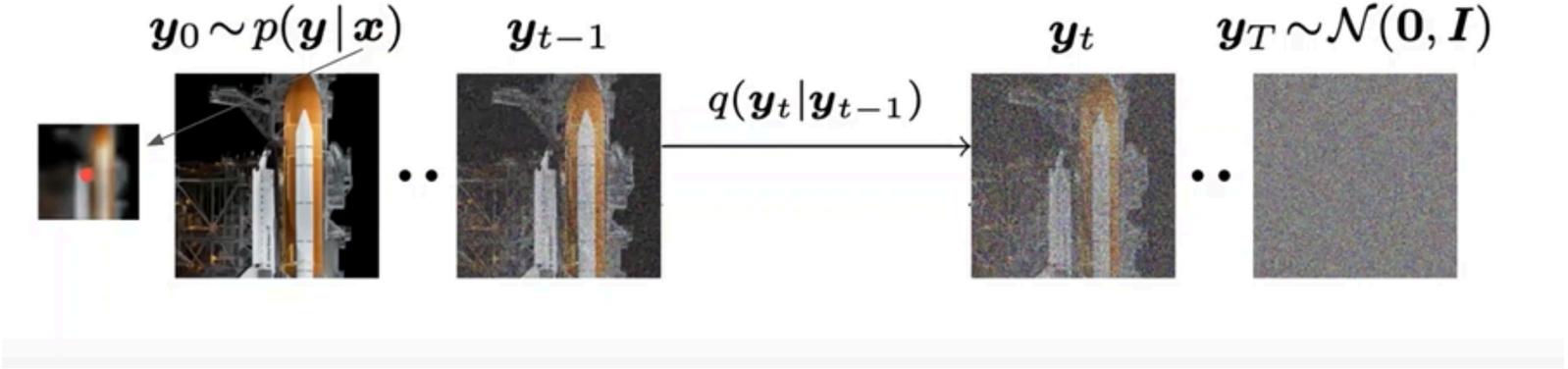
$$\gamma_t = \prod_{i=1}^{t} \alpha_i$$

$$q(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_0, \boldsymbol{y}_t) = \mathcal{N}(\boldsymbol{y}_{t-1} \mid \boldsymbol{\mu}, \sigma^2 \boldsymbol{I})$$

$$\boldsymbol{\mu} = \frac{\sqrt{\gamma_{t-1}}\,(1-\alpha_t)}{1-\gamma_t}\,\boldsymbol{y}_0 + \frac{\sqrt{\alpha_t}\,(1-\gamma_{t-1})}{1-\gamma_t}\,\boldsymbol{y}_t$$

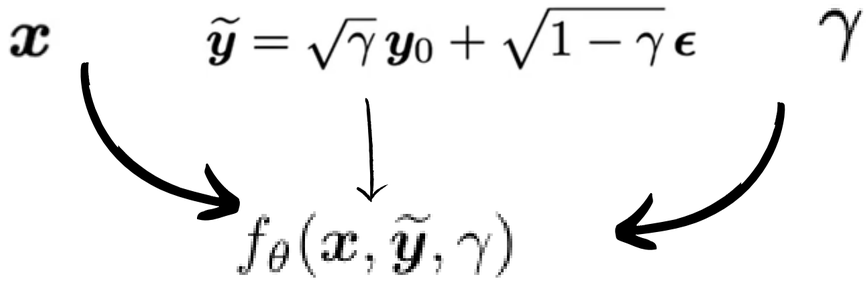
$$\sigma^2 = \frac{(1-\gamma_{t-1})(1-\alpha_t)}{1-\gamma_t}\,.$$

Conditional Denoising Diffusion Model



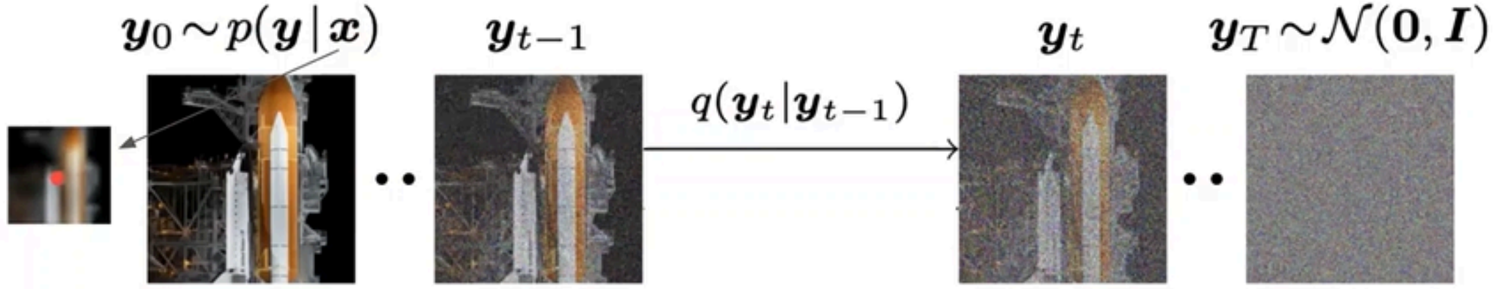
We learn the reverse chain using a neural denoising model f that takes as input a source image and a noisy target image and estimates the noise

To help reverse the diffusionprocess,we take advantage of additional side information in the form of source image and and a noisy target image y, takes as input the sufficient statistics for the variance of the noise



$$\mathbb{E}_{(x,y)} \mathbb{E}_{\epsilon,\gamma} \left\| f_\theta(x, \underbrace{\sqrt{\gamma}\, y_0 + \sqrt{1-\gamma}\, \epsilon}_{\tilde{y}}, \gamma) - \epsilon \right\|_p^p,$$

Conditional Denoising Diffusion Model



$$
\begin{aligned}
p_\theta(\boldsymbol{y}_{0:T}|\boldsymbol{x}) &= p(\boldsymbol{y}_T)\prod_{t=1}^{T} p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x}) \quad (7)\\
p(\boldsymbol{y}_T) &= \mathcal{N}(\boldsymbol{y}_T \mid \mathbf{0}, \boldsymbol{I}) \quad (8)\\
p_\theta(\boldsymbol{y}_{t-1}|\boldsymbol{y}_t, \boldsymbol{x}) &= \mathcal{N}(\boldsymbol{y}_{t-1} \mid \mu_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t), \sigma_t^2 \boldsymbol{I}) \; . \quad (9)
\end{aligned}
$$

$$
\hat{\boldsymbol{y}}_0 = \frac{1}{\sqrt{\gamma_t}}\left(\boldsymbol{y}_t - \sqrt{1-\gamma_t}\, f_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t)\right) \; .
$$

$$
\mu_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t) = \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t)\right)
$$

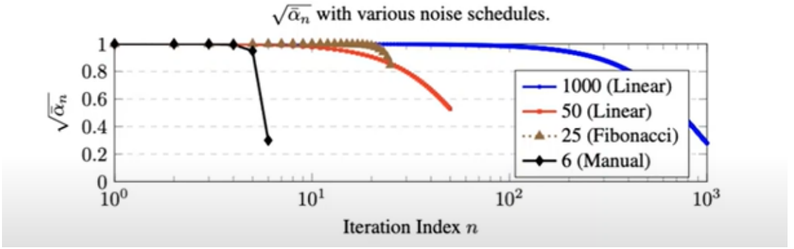
$$
\boldsymbol{y}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}\left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t)\right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t
$$

Noise Scheduling

β_t controls noise level at each step of forward diffusion process. It is typically chosen to increase gradually, ensuring a smooth transition from data to noise

$\bar{\alpha}_t$ represents retention rate of the data's original signal at each step. It decreases gradually, reflecting increasing noise level

γ_t indicates overall retention rate up to step t -> crucial for computing mean of reverse process distribution



The noise shedule is insipired by WaveGrad.

The model conditions on directly on gamma (vs t), which allows us flexibility in choosing number of diffusion steps, and the noise schedule during inference. This has been demonstrated to work well for speech synthesis but has not been explored for images.

Prior work of diffusion models require 1-2k dif fusion steps during inference, making generation slow for large target resolution tasks.

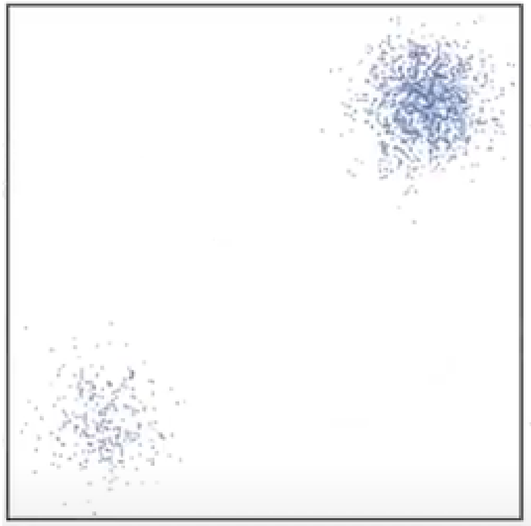
We can chose number of steps for inference flexibly without training the multiple models which makes this cost effective

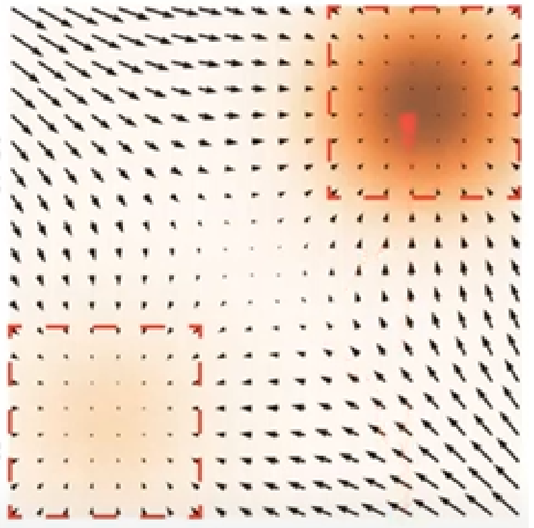
$p(\gamma) = \sum_{t=1}^{T} \frac{1}{T} U(\gamma_{t-1}, \gamma_t)$ piece wise distribution

$t \sim \{0, ..., T\}$ T = 2000

$U(\gamma_{t-1}, \gamma_t)$

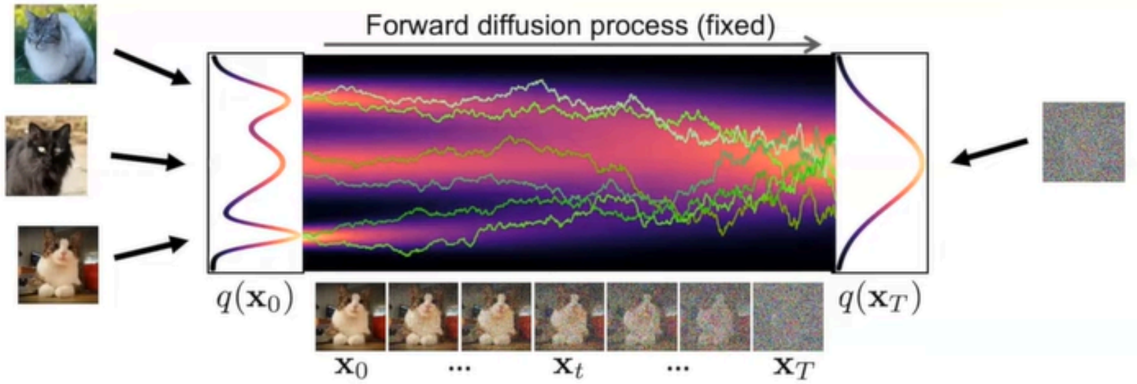
$$\boldsymbol{y}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t) \right) + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t \ ,$$





An instresting view corresponding Langevin dynamics

The Generative Reverse Stochastic Differential Equation



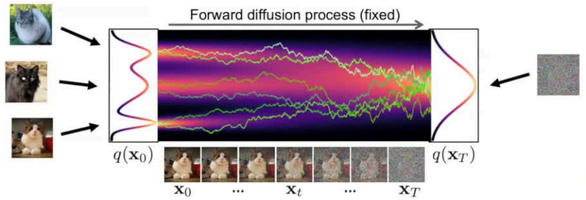
Forward Diffusion SDE:

$$\mathrm{d}\mathbf{x}_t = -\frac{1}{2}\beta(t)\mathbf{x}_t\,\mathrm{d}t + \sqrt{\beta(t)}\,\mathrm{d}\boldsymbol{\omega}_t$$

Reverse Generative Diffusion SDE:

$$\mathrm{d}\mathbf{x}_t = \overbrace{\left[-\frac{1}{2}\beta(t)\mathbf{x}_t - \beta(t)\underbrace{\nabla_{\mathbf{x}_t}\log q_t(\mathbf{x}_t)}_{\text{“Score Function”}}\right]}^{\text{drift term}}\mathrm{d}t + \overbrace{\sqrt{\beta(t)}\,\mathrm{d}\bar{\boldsymbol{\omega}}_t}^{\text{diffusion term}}$$

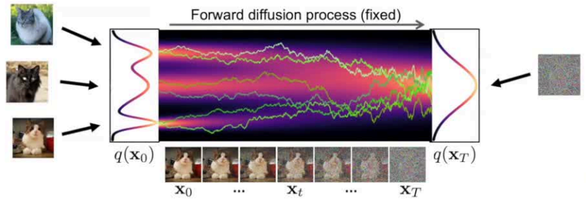
An instresting view



- Naïve idea, learn model for the score function by direct regression?

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim\mathcal{U}(0,T)} \mathbb{E}_{\mathbf{x}_t\sim q_t(\mathbf{x}_t)} ||\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)||_2^2$$

diffusion time t | diffused data $\mathbf{x}_t$ | neural network | score of diffused data (marginal)



- Naïve idea, learn model for the score function by direct regression?

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim\mathcal{U}(0,T)} \mathbb{E}_{\mathbf{x}_t\sim q_t(\mathbf{x}_t)} ||\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t)||_2^2$$

diffusion time t | diffused data $\mathbf{x}_t$ | neural network | score of diffused data (marginal)

Denoising Score Matching:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{t\sim\mathcal{U}(0,T)} \mathbb{E}_{\mathbf{x}_0\sim q_0(\mathbf{x}_0)} \mathbb{E}_{\mathbf{x}_t\sim q_t(\mathbf{x}_t|\mathbf{x}_0)} ||\mathbf{s}_{\boldsymbol{\theta}}(\mathbf{x}_t, t) - \nabla_{\mathbf{x}_t} \log q_t(\mathbf{x}_t|\mathbf{x}_0)||_2^2$$

$$q(\widetilde{\boldsymbol{y}}|\boldsymbol{y}_0, \gamma) = \mathcal{N}(\widetilde{\boldsymbol{y}} \mid \sqrt{\gamma}\boldsymbol{y}_0, 1-\gamma)$$

$$\frac{\mathrm{d}\log q(\widetilde{\boldsymbol{y}} \mid \boldsymbol{y}_0, \gamma)}{\mathrm{d}\widetilde{\boldsymbol{y}}} = -\frac{\widetilde{\boldsymbol{y}} - \sqrt{\gamma}\boldsymbol{y}_0}{\sqrt{1-\gamma}} = -\boldsymbol{\epsilon}$$

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})} \mathbb{E}_{\boldsymbol{\epsilon},\gamma} \left\| f_\theta(\boldsymbol{x}, \underbrace{\sqrt{\gamma}\,\boldsymbol{y}_0 + \sqrt{1-\gamma}\,\boldsymbol{\epsilon}}_{\widetilde{\boldsymbol{y}}}, \gamma) - \boldsymbol{\epsilon} \right\|_p^p,$$

EXPERIMENTATION

Algorithm 1 Training a denoising model f_θ

1: **repeat**
2: $(\boldsymbol{x}, \boldsymbol{y}_0) \sim p(\boldsymbol{x}, \boldsymbol{y})$
3: $\gamma \sim p(\gamma)$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take a gradient descent step on
$\nabla_\theta \left\| f_\theta(\boldsymbol{x}, \sqrt{\gamma}\boldsymbol{y}_0 + \sqrt{1-\gamma}\boldsymbol{\epsilon}, \gamma) - \boldsymbol{\epsilon} \right\|_p^p$
6: **until** converged

Algorithm 2 Inference in T iterative refinement steps

1: $\boldsymbol{y}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \ldots, 1$ **do**
3: $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\boldsymbol{z} = \mathbf{0}$
4: $\boldsymbol{y}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\boldsymbol{y}_t - \frac{1-\alpha_t}{\sqrt{1-\gamma_t}} f_\theta(\boldsymbol{x}, \boldsymbol{y}_t, \gamma_t) \right) + \sqrt{1-\alpha_t}\boldsymbol{z}$
5: **end for**
6: **return** $\boldsymbol{y}_0$

compare to a regression baseline model that shares the same architecture as SR3, but is trained with a MSE loss.

- Face super-resolution at $16\times16 \rightarrow 128\times128$ and $64\times64 \rightarrow 512 \times 1512$ trained on FFHQ and evaluated on CelebA-HQ.
- Natural image super-resolution at $64 \times 64 \rightarrow 256 \times 256$ pixels on ImageNet [43].
- Unconditional 1024×1024 face generation by a cascade of 3 models, and class-conditional 256×256 ImageNet image generation by a cascade of 2 models.

TRAINING DETAILS

Training Steps and Batch Size:

All models are trained for 1M steps with a batch size of 256.

Checkpoint Selection:
Regression models: Checkpoints are selected based on peak PSNR on the validation set.
SR3 models: The latest checkpoint is always used without checkpoint selection.

Optimizer and Learning Rate Schedule:
Adam optimizer with
Linear warmup for the first 10k steps.

Fixed learning rates:
1e-4 for SR3 models.
1e-5 for regression models.

TRAINING DETAILS

Training Steps and Batch Size:

All models are trained for 1M steps with a batch size of 256.

Checkpoint Selection:
Regression models: Checkpoints are selected based on peak PSNR on the validation set.
SR3 models: The latest checkpoint is always used without checkpoint selection.

Optimizer and Learning Rate Schedule:
Adam optimizer with
Linear warmup for the first 10k steps.

Fixed learning rates:
1e-4 for SR3 models.
1e-5 for regression models.

RESULTS

Figure 4: Results of a SR3 model (64×64 → 512×512), trained on FFHQ, and applied to images outside of the training set, enlarged patches to show finer details. Additional results are shown in Appendix C.1 and C.2.

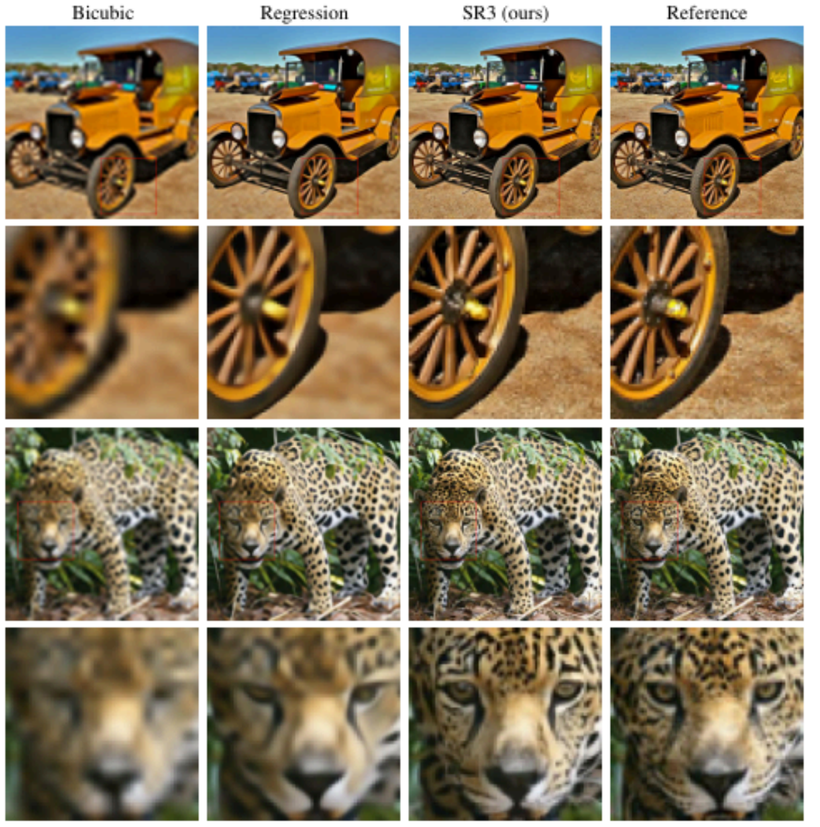


Figure 3: Results of a SR3 model (64×64 → 256×256), trained on ImageNet and evaluated on two ImageNet test images. For each we also show an enlarged patch in which finer details are more apparent. Additional samples are shown in Appendix C.3 and C.4.

Cascaded Face Generation 1024×1024

(64×64) (256×256) (1024×1024)

Class Conditional ImageNet Samples 256×256

Metric	PULSE [28]	FSRGAN [7]	Regression	SR3
PSNR ↑	16.88	23.01	**23.96**	23.04
SSIM ↑	0.44	0.62	**0.69**	0.65
Consistency ↓	161.1	33.8	2.71	**2.68**

Table 1: PSNR & SSIM on $16\times16 \rightarrow 128\times128$ face super-resolution. Consistency measures MSE ($\times10^{-5}$) between the low-resolution inputs and the down-sampled super-resolution outputs.

Model	FID ↓	IS ↑	PSNR ↑	SSIM ↑
Reference	1.9	240.8	-	-
Regression	15.2	121.1	**27.9**	**0.801**
SR3	**5.2**	**180.1**	26.4	0.762

Table 2: Performance comparison between SR3 and Regression baseline on natural image super-resolution using standard metrics computed on the ImageNet validation set.

Method	Top-1 Error	Top-5 Error
Baseline	0.252	0.080
DRCN [22]	0.477	0.242
FSRCNN [13]	0.437	0.196
PsyCo [35]	0.454	0.224
ENet-E [44]	0.449	0.214
RCAN [64]	0.393	0.167
Regression	0.383	0.173
SR3	**0.317**	**0.120**

Table 3: Comparison of classification accuracy scores for $4\times$ natural image super-resolution on the first 1K images from the ImageNet Validation set.

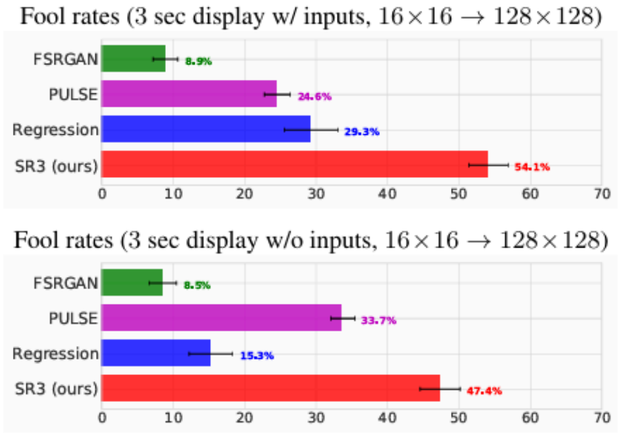


Figure 6: Face super-resolution human fool rates (higher is bet ter, photo-realistic samples yield a fool rate of 50%). Outputs o 4 models are compared against ground truth. (top) Subjects ar shown low-resolution inputs. (bottom) Inputs are not shown.

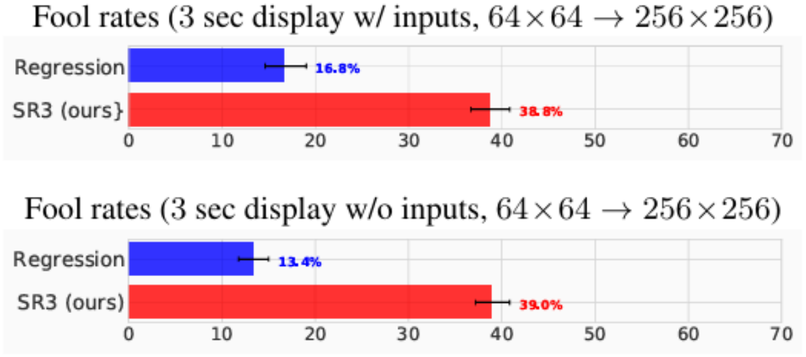


Figure 7: ImageNet super-resolution fool rates (higher is better, photo-realistic samples yield a fool rate of 50%). SR3 and Regression outputs are compared against ground truth. (top) Subjects are shown low-resolution inputs. (bottom) Inputs are not shown.

CLASS CONDITIONAL

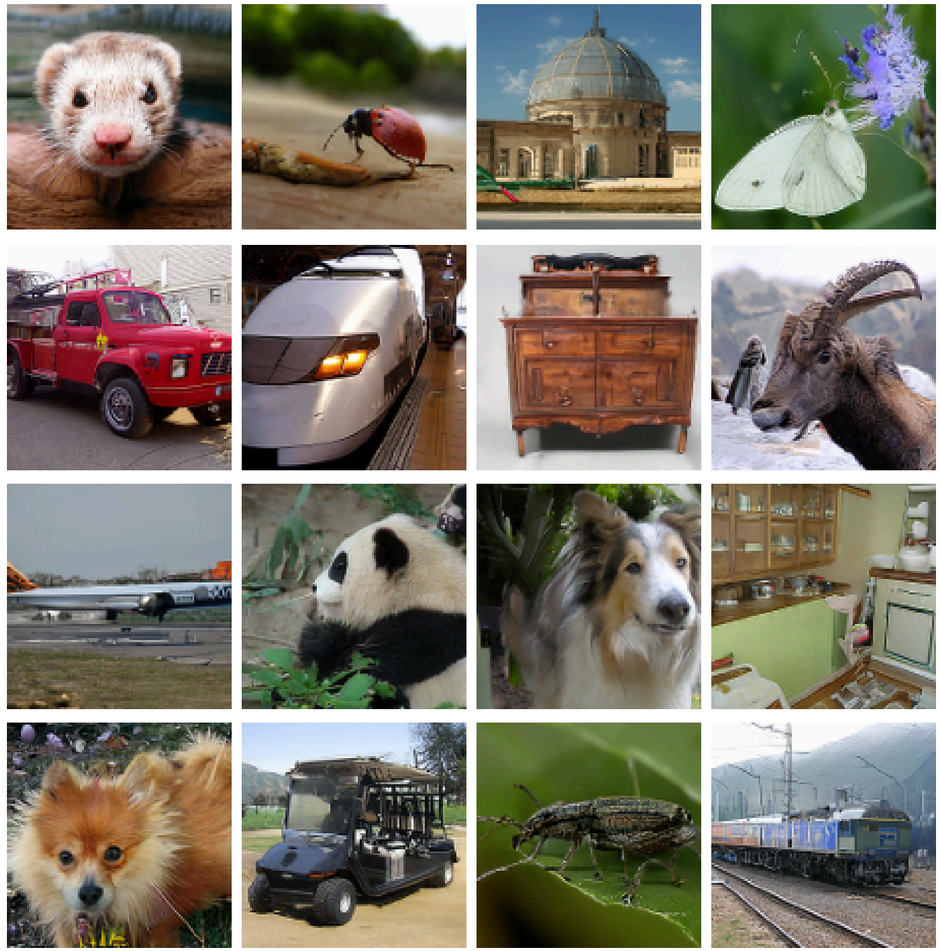


Figure 9: Synthetic 256×256 ImageNet images. We first draw a random label, then sample a 64×64 image from a class-conditional diffusion model, and apply a 4× SR3 model to obtain 256×256 images. Additional samples in Appendix C.10 and C.11.

Model	FID-50k
Prior Work	
VQ-VAE-2 [39]	38.1
BigGAN (Truncation 1.0) [4]	7.4
BigGAN (Truncation 1.5) [4]	11.8
Our Work	
SR3 (Two Stage)	11.3

Table 4: FID scores for class-conditional 256×256 ImageNet.

Model	FID-50k
Training with Augmentation	
SR3	13.1
SR3 (w/ Gaussian Blur)	11.3
Objective L_p Norm	
SR3 (L_2)	11.8
SR3 (L_1)	11.3

Limitations

We also ob served the model to generate very continuous skin texture in face super-resolution, dropping moles, pimples and pierc ings found in the reference.

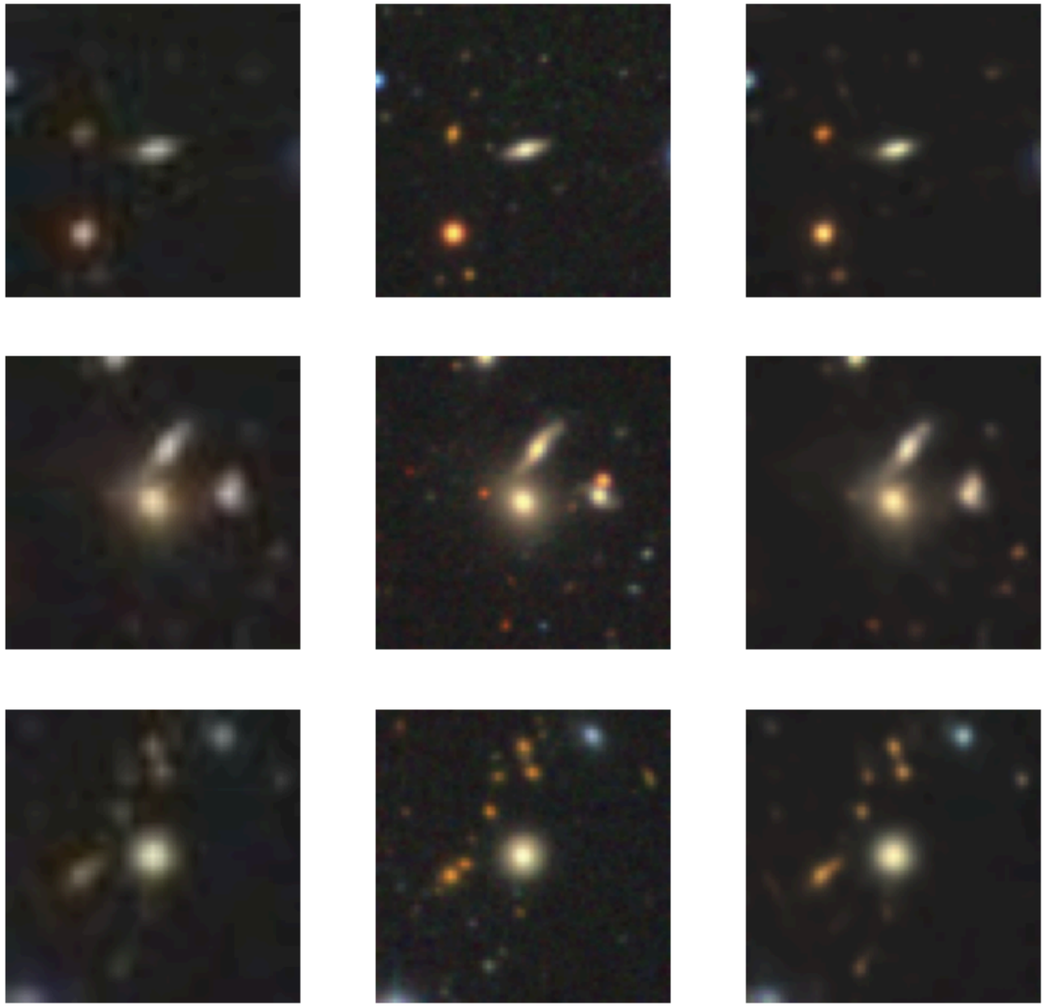
References

Image Super-Resolution via Iterative Refinement | IEEE Journals & Magazine | IEEE Xplore

Tutorial on Denoising Diffusion-based Generative Modeling: Foundations and Applications
(CVPR 2022 tutorial)

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Moham mad Norouzi, and William Chan.
WaveGrad: Estimating Gradients for Waveform Generation. In ICLR, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffu sion probabilistic models. In NeurIPS,
2020.



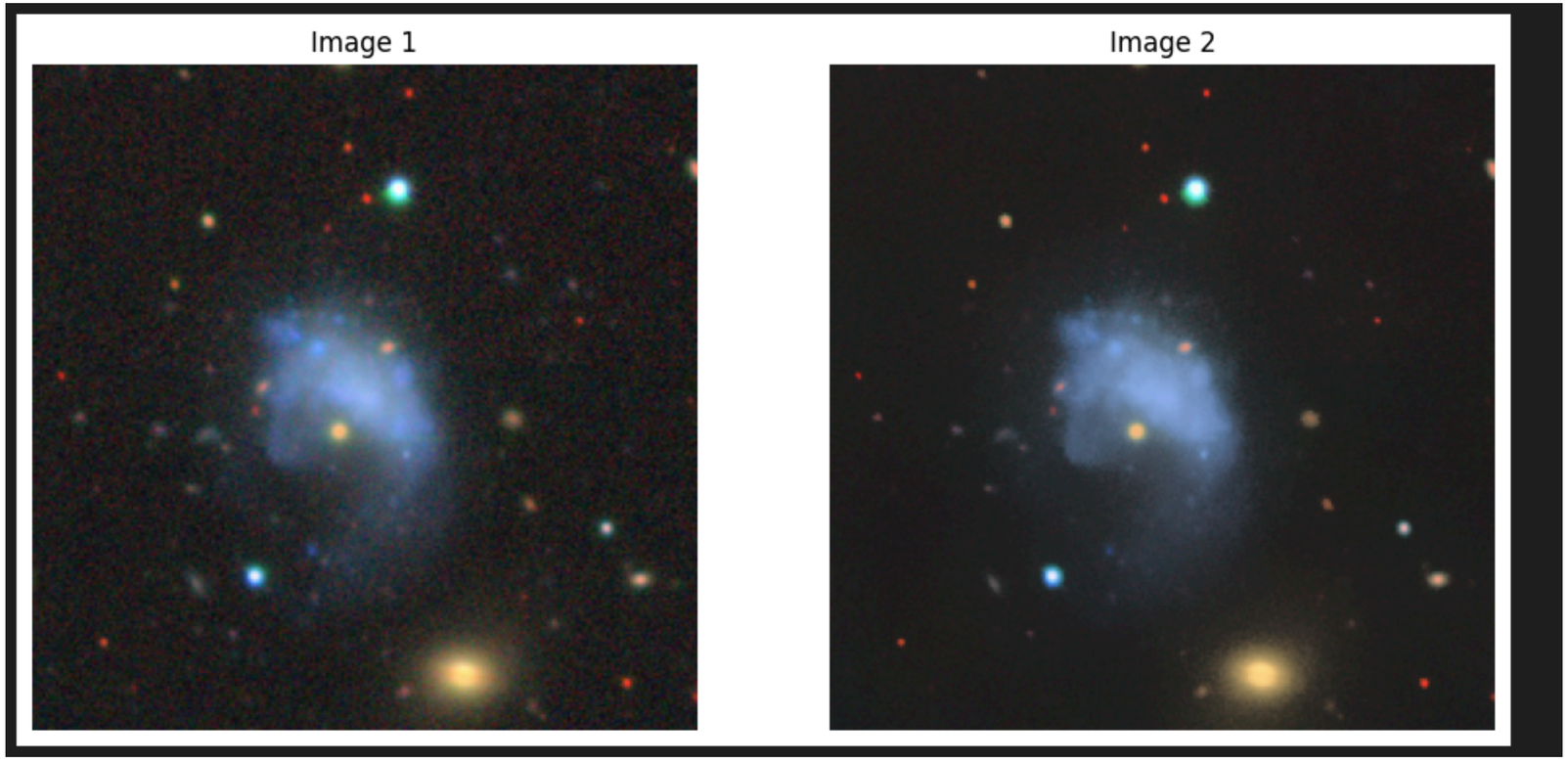
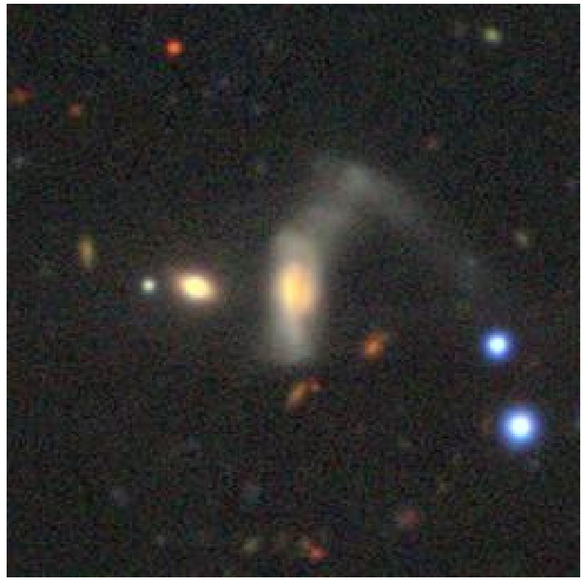
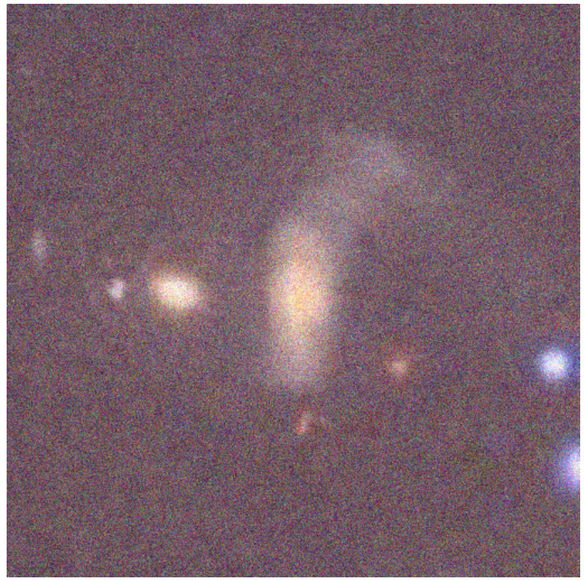
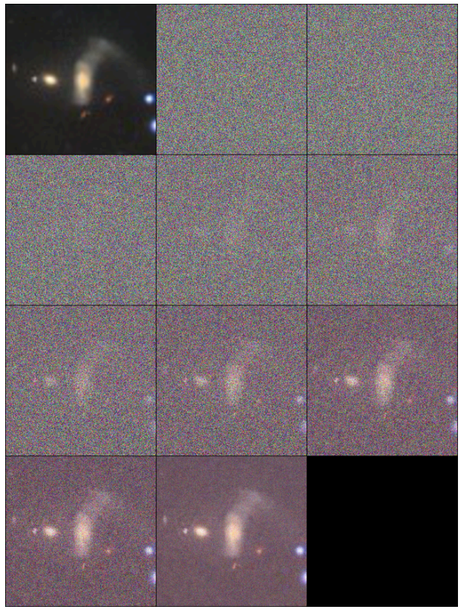
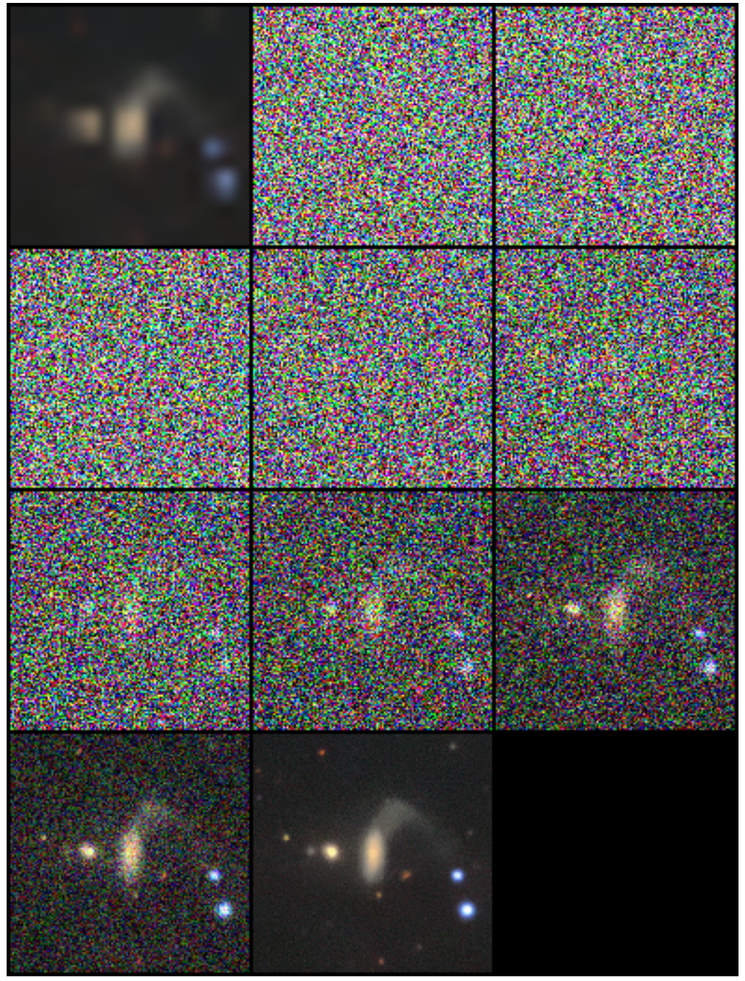


Image 1
Image 2

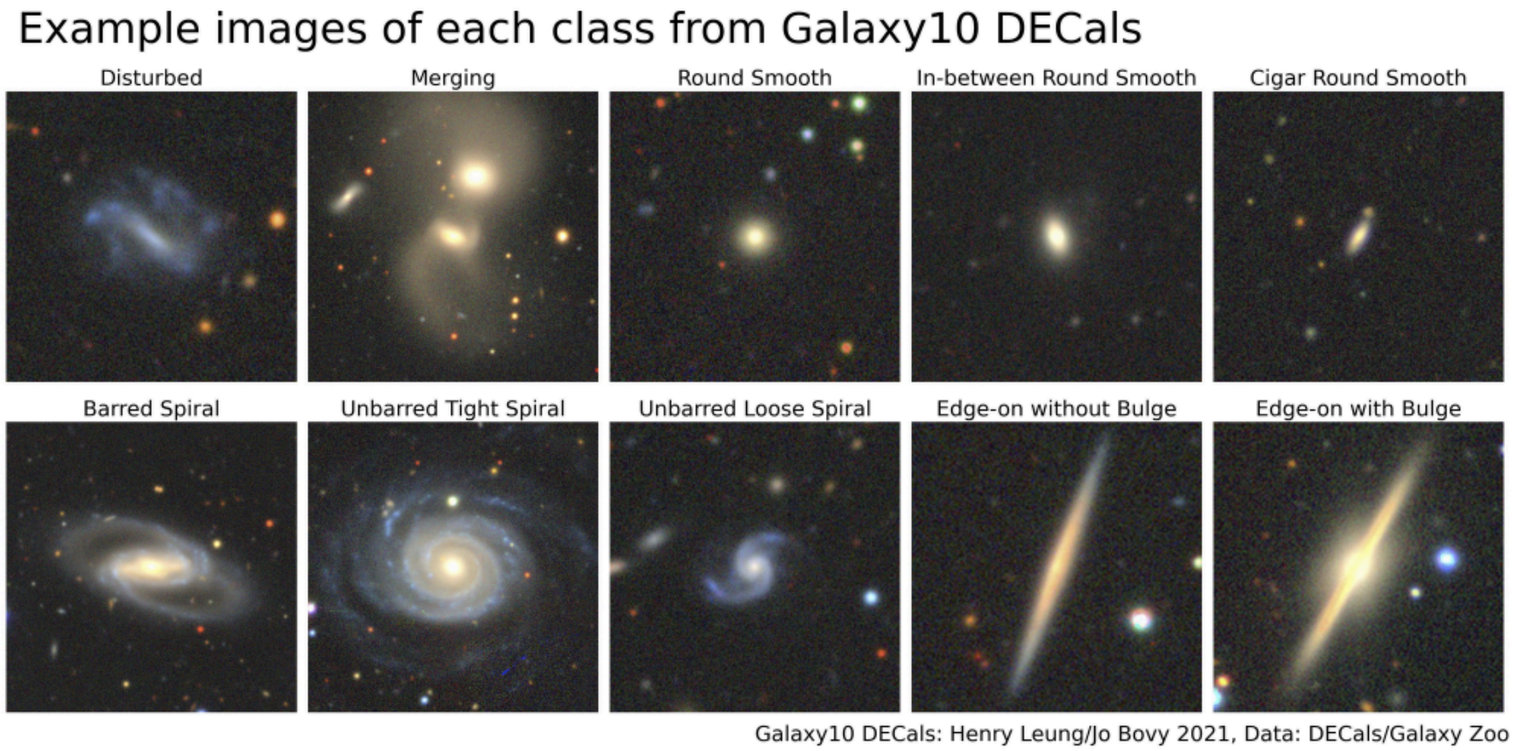




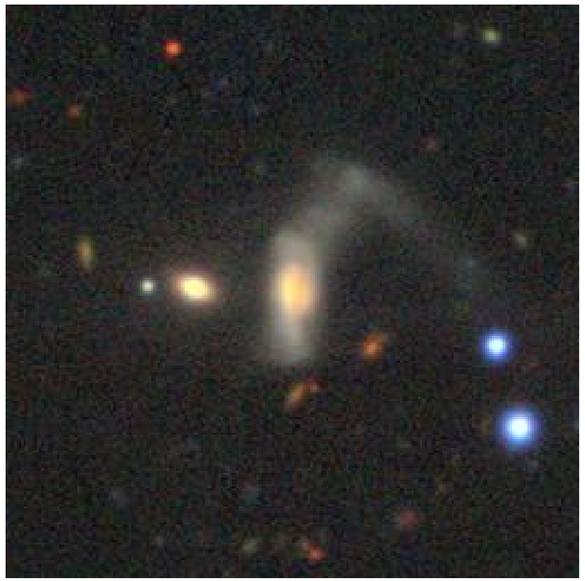


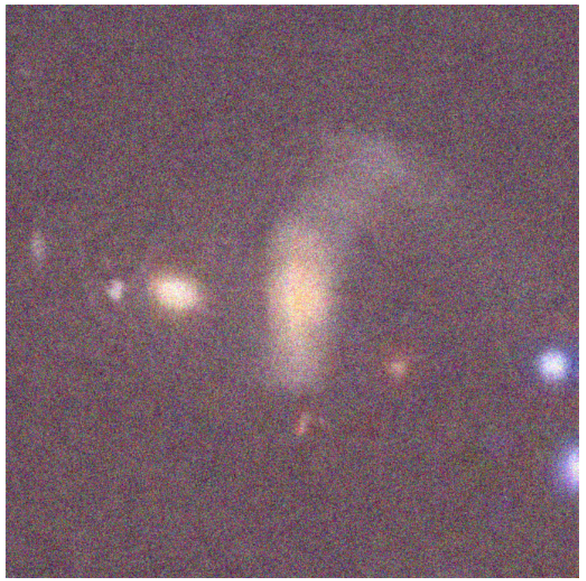


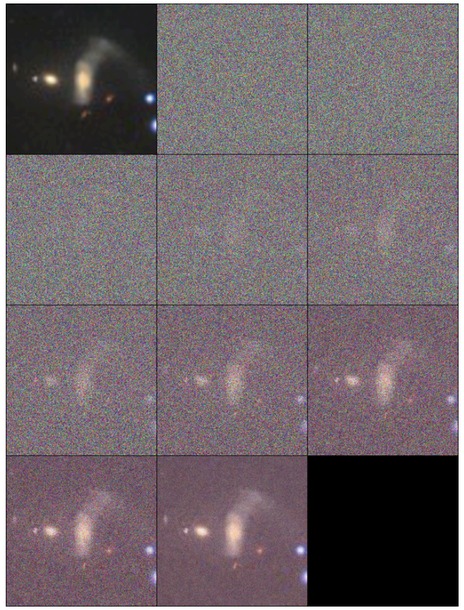
dataset


Example images of each class from Galaxy10 DECals
Disturbed
Merging
Round Smooth
In-between Round Smooth
Cigar Round Smooth
Barred Spiral
Unbarred Tight Spiral
Unbarred Loose Spiral
Edge-on without Bulge
Edge-on with Bulge
Galaxy10 DECals: Henry Leung/Jo Bovy 2021, Data: DECals/Galaxy Zoo

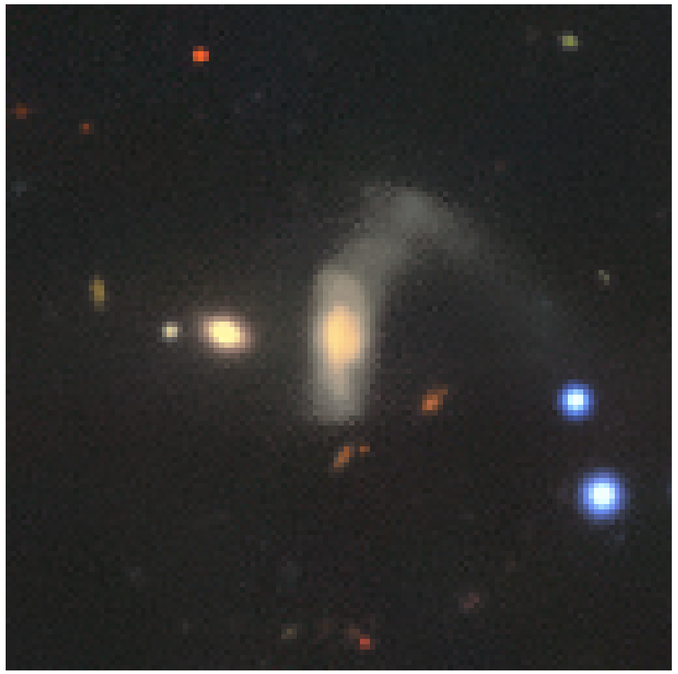
pretrained with noise

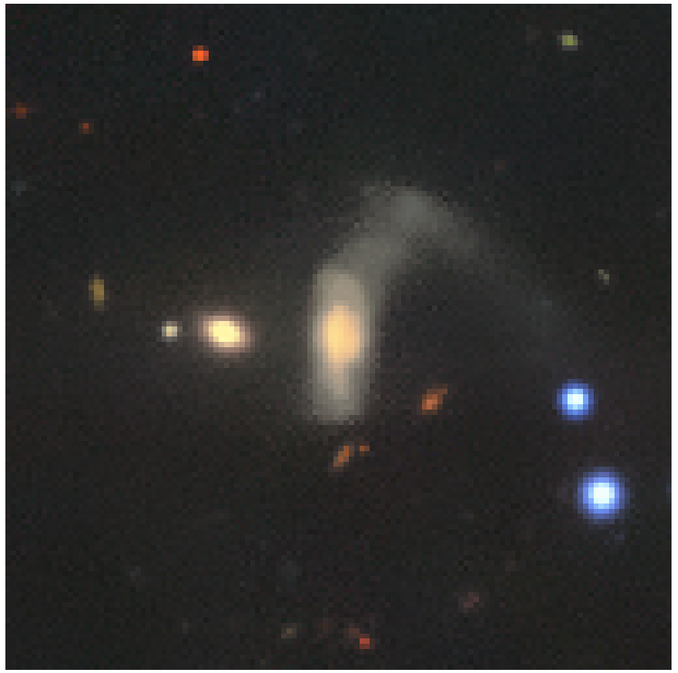


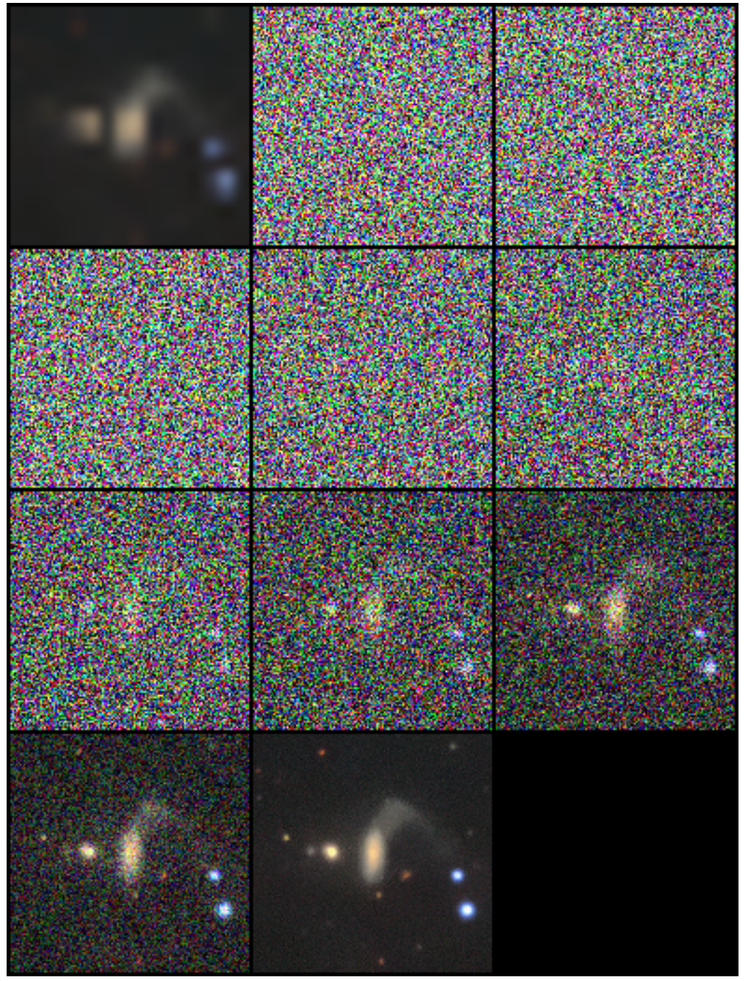




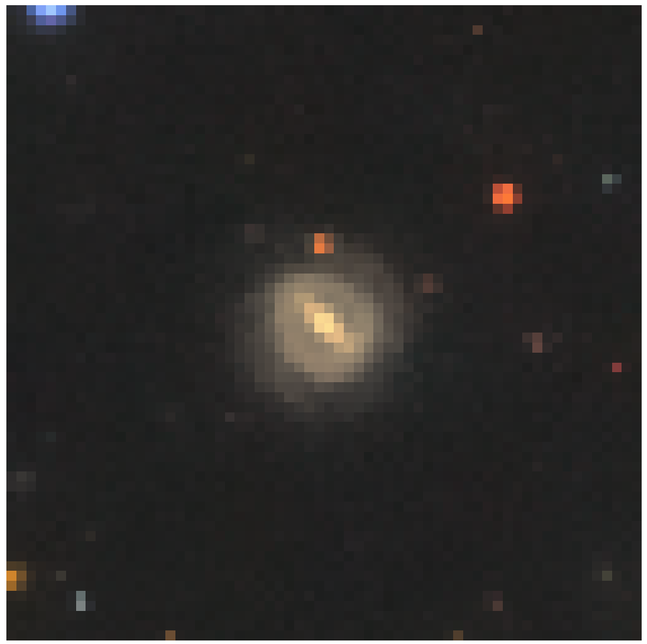
pretrained with bilateral



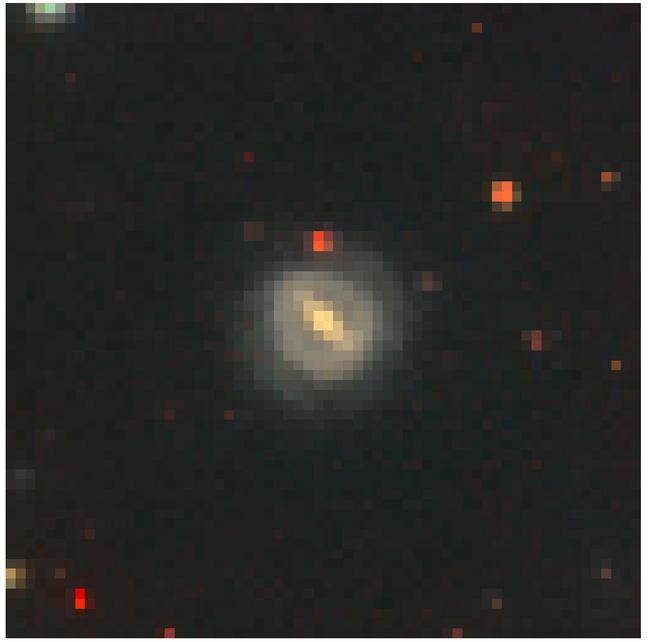




original

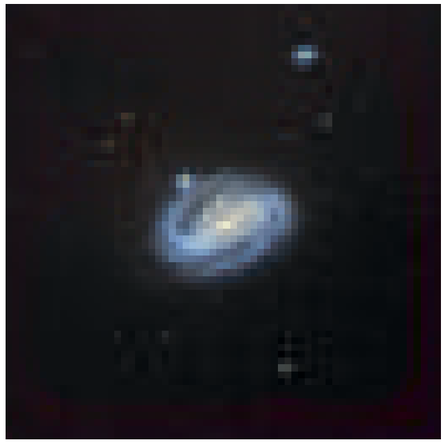


Colorized

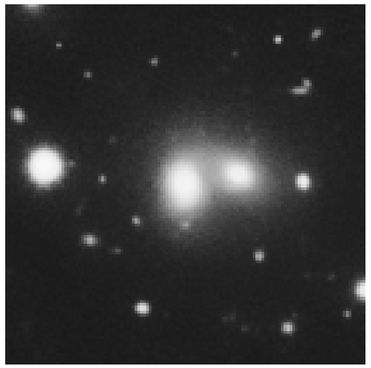


Original

Colorized



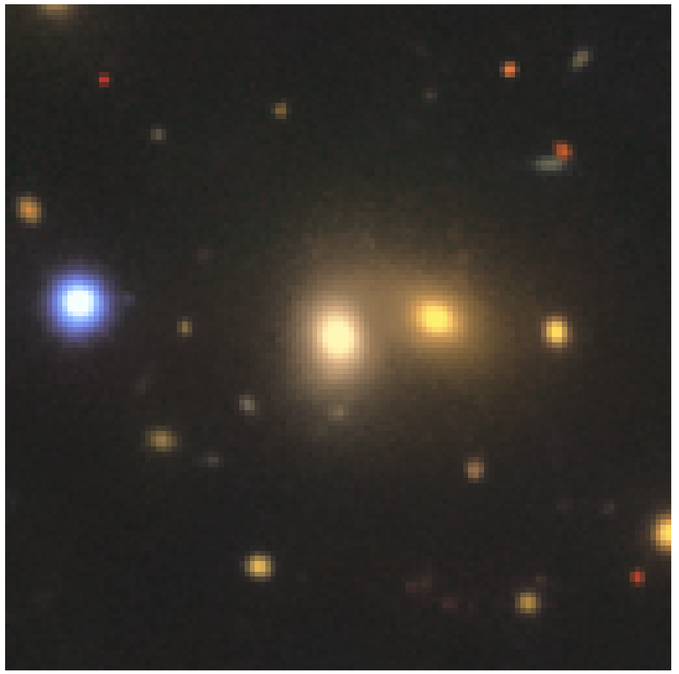
combined



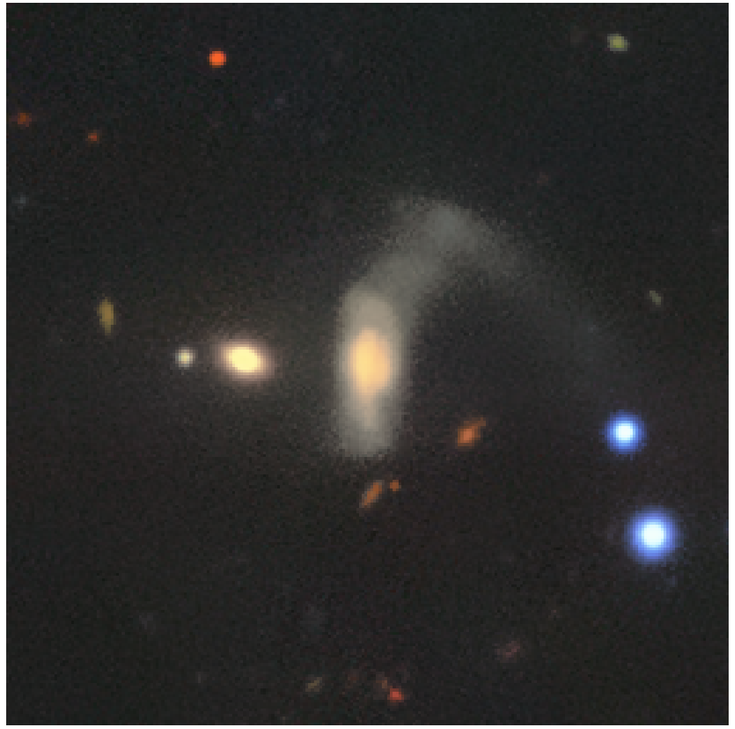
black and white

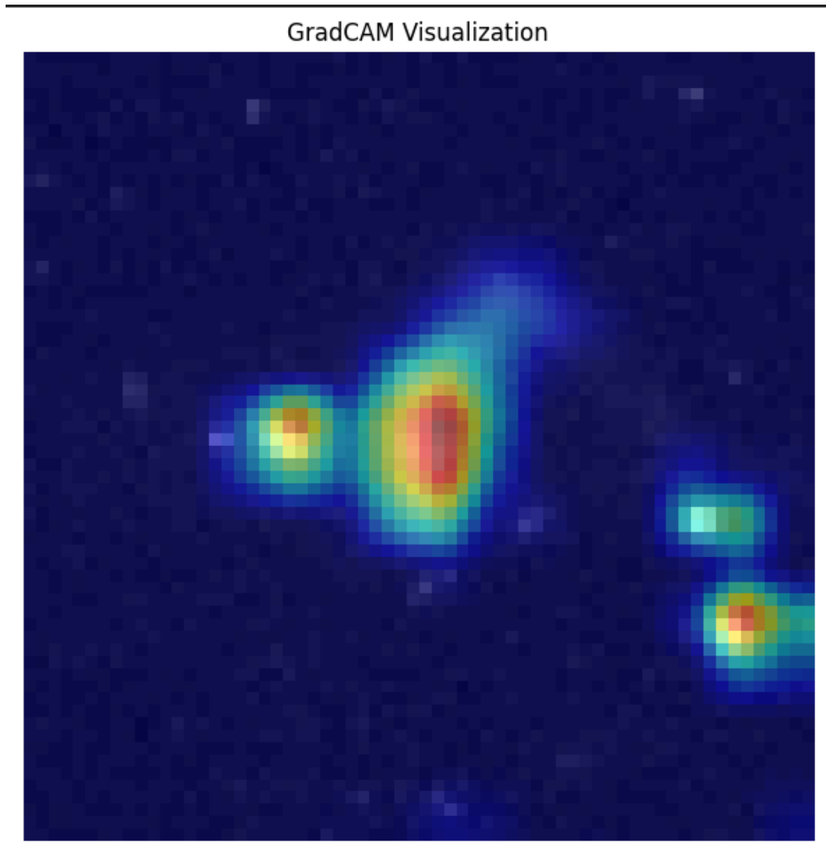
colored

SR

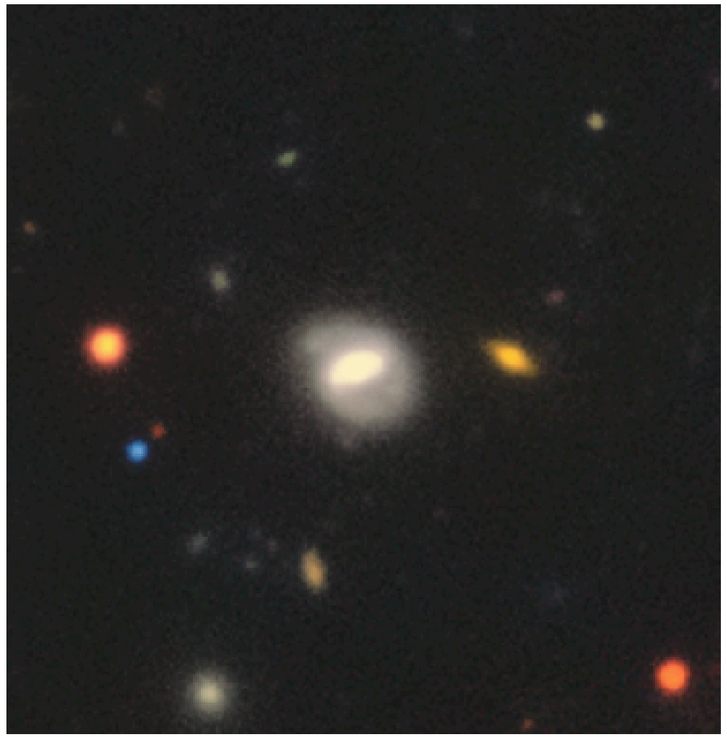


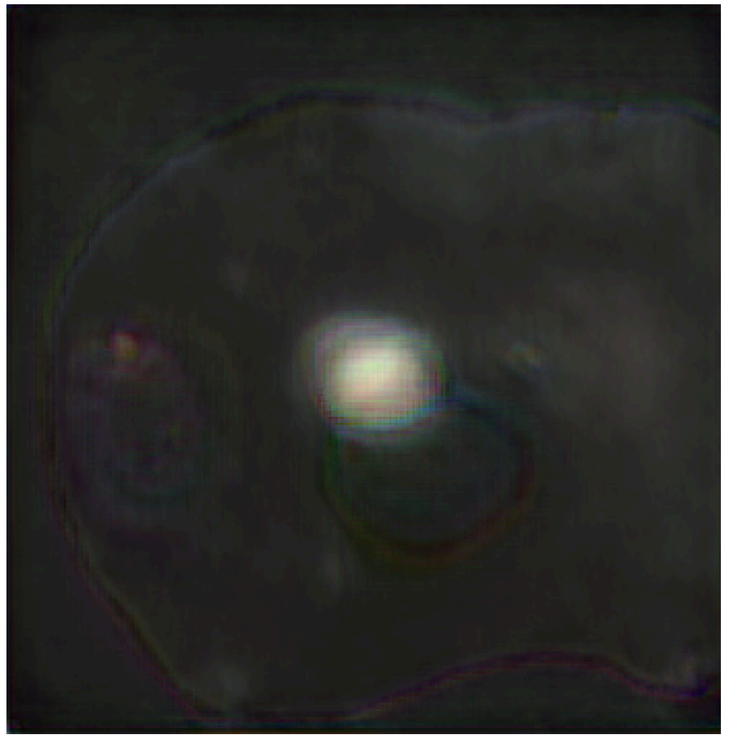
REAL

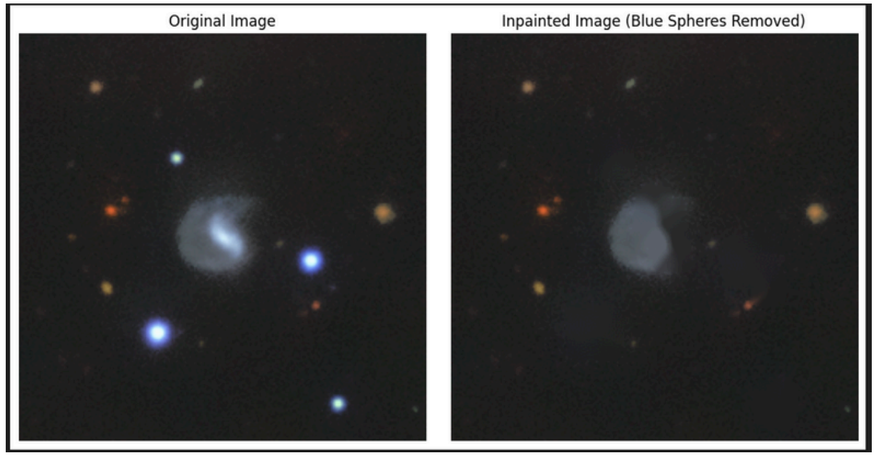


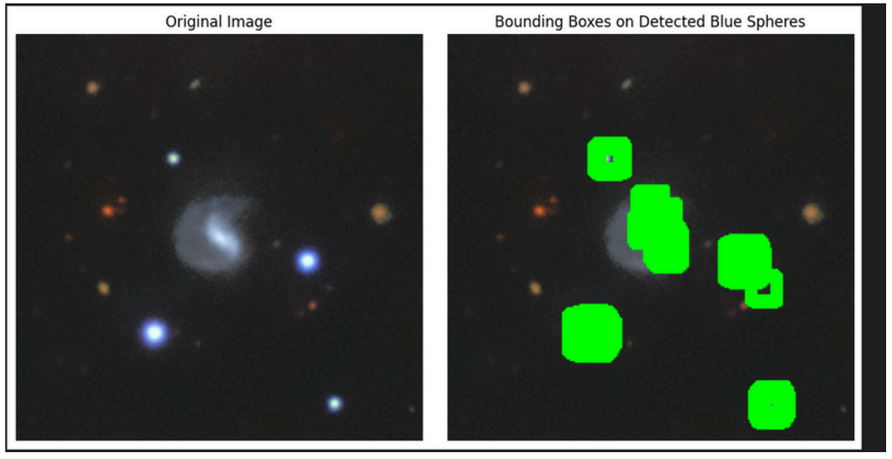

GradCAM Visualization

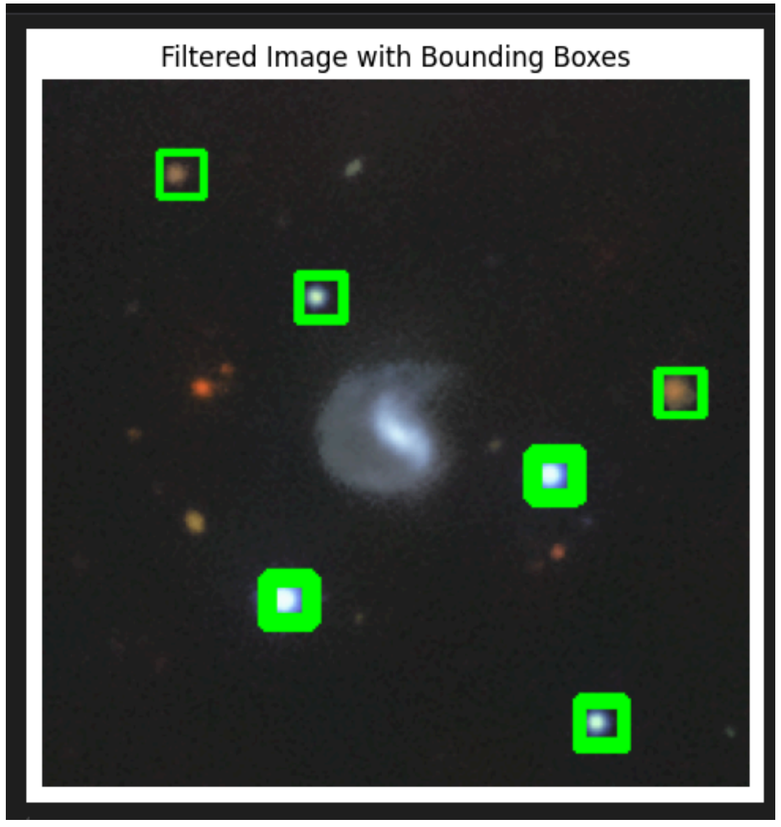
unet autoencoder






Original Image
Inpainted Image (Blue Spheres Removed)


Original Image
Bounding Boxes on Detected Blue Spheres


Filtered Image with Bounding Boxes