ELEG404/604: Imaging & Deep Learning

Gonzalo R. Arce

Department of Electrical and Computer Engineering
University of Delaware

Image Deblurring

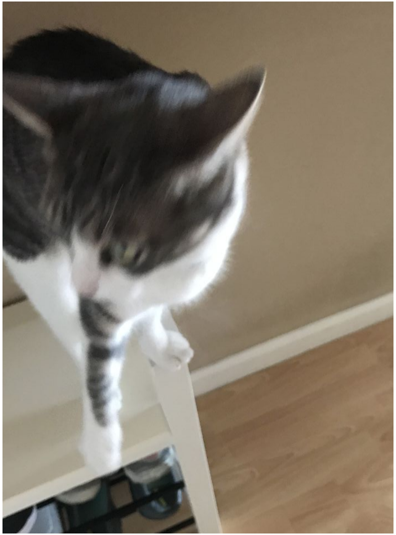
Blur in images
Example - Lens blur

Blur in images

Example - Out-of-focus/DOF

Blur in images

Example - Object motion

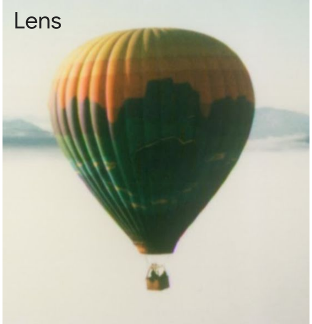


Blur in images

Example - Camera motion

Camera Shake - Camera motion Blur

- Low-light environments (but not only).


Lens


Camera-shake


Object-motion


Out-of-focus


limited DOF


Camera-motion

Blur in images

Different sources of blur with very different properties

1. Lens Blur (mostly isotropic, "mostly" spatially invariant)
2. Out-of-focus blur / DOF (mostly isotropic, might change spatially DOF)
3. Object motion (non-isotropic, abruptly changes spatially due to the object moving)
4. Camera shake/motion (non-isotropic, may vary spatially but changes smoothly)

Challenges: Removing blur in arbitrary images

- Arbitrary Images may have suffered a lot of *unknown* processing:
  - Demosaicing, Denoising
  - Multi-frame fusion
  - Gamma correction, non-linear CRF
  - Compression, resizing
- Don't have a perfect model of the blurring process.


blurry


Deblurred
RODEO

Image Deblurring as an *Inverse Problem*

- Many works from 80's (and before!)
- Classical methods ***model*** degradation and space of solutions (high-quality images)
- Current trend: Data-based methods (Data-Model), Deep Learning
- Image Deblurring → Inverse Problem

Image Deblurring as an *Inverse Problem*

$$\mathbf{y} = A(\mathbf{x}) + \mathbf{n}$$

What we have:
"The measurements"

Forward model
Known/Unknown (sampling, blur, compression,...)

What we want:
the clean signal

Noise in the measurements
Unknown (statistically known maybe)

A: Convolution with blur kernel

Image Deblurring as an *Inverse Problem*

$$\mathbf{y} = A(\mathbf{x}) + \mathbf{n}$$

What we have:
"The measurements"

Forward model
Known/Unknown (sampling, blur, compression,...)

What we want:
the clean signal

Noise in the measurements
Unknown (statistically known maybe)

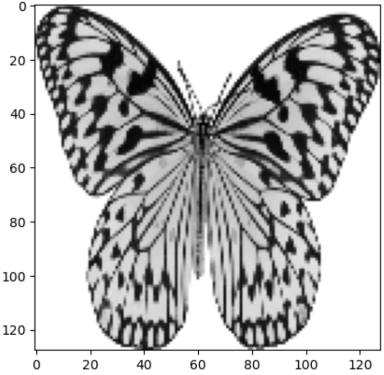
We need to **invert A** but:

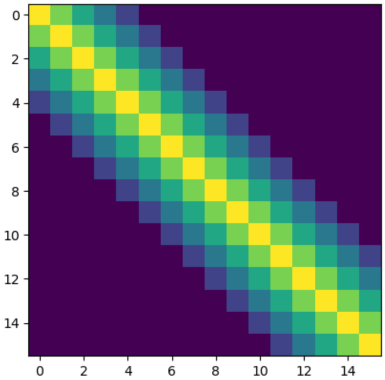
- **A** might be singular
- There's noise **n**

Many possible solutions **x** leading to the same (approx) **y**.

Problem Setup

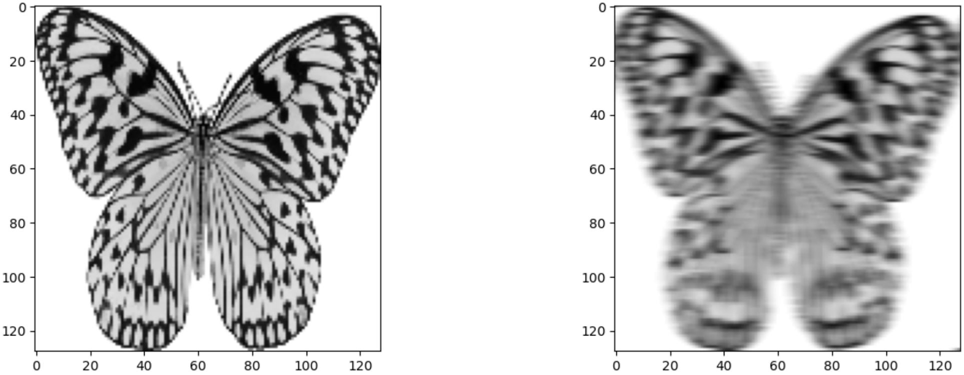
- Let $\mathbf{X} \in \mathbb{R}^{n \times m}$ be a sharp gray scale image.
- Let $\mathbf{x} \in \mathbb{R}^{nm}$ be the reindexed version of the sharp image as a single column vector.
- Consider the linear operator $\mathbf{A} \in \mathbb{R}^{nm \times nm}$ defined as a square Toeplitz matrix with diagonal operator $[1/15, 2/15, 3/15, 4/15, 3/15, 2/15, 1/15]$.





Problem Setup

- Let $\mathbf{y}_b = \mathbf{A}\mathbf{x}$ be the vector version of the blurry image.
- Notice that given the structure of $\mathbf{A}$, this operation is averaging adjacent pixels in one axis of the original image.

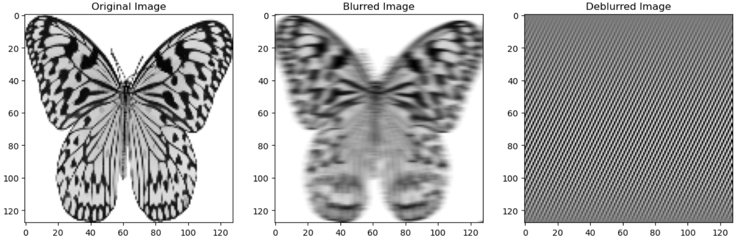


Inverse Problem

- What if we were given the blurry image $\mathbf{y}_b$?
- One might be tempted to solve the linear system of equations as:

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}_b,$$

thus recovering the sharp image $\mathbf{x}$.
- However, $\mathbf{A}$ might not be invertible or be very high dimensional (computationally expensive to invert).



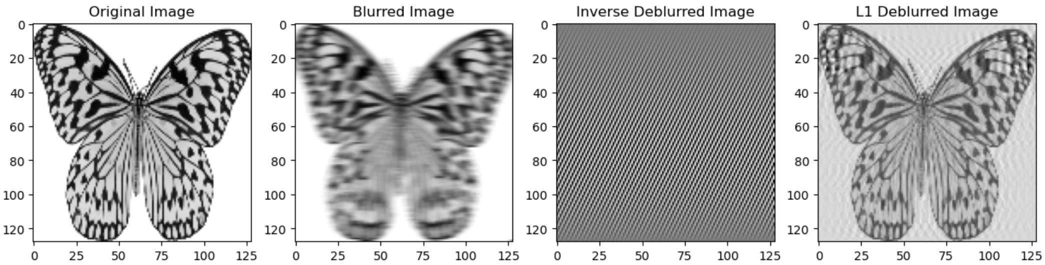
Inverse Problem - Solution

- A different approach to this problem would be to solve it as an optimization problem:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$$

- Assume the solution is sparse in some domain:

$$\min_{\theta} \|\mathbf{y} - \mathbf{A}\mathbf{\Phi}\theta\|_2^2 + \lambda\|\theta\|_1^2$$



Variational Formulation (Model based)

$$(\mathbf{x}, A) = \arg\min_{\mathbf{x},A} \boxed{||A(\mathbf{x}) - \mathbf{y}||^2} + \lambda \boxed{R_1(\mathbf{x})} + \mu \boxed{R_2(A)}$$

Data fitting (blind)
Clean signal matches the observation

Prior information on x
(Regularity, lots of papers...)

Prior information on A
(e.g., unknown motion blur)

- **Optimization Problem**: given **y** recover **x** (and **A**)
- **R1, R2**: can be non-convex
- Iterative methods (GD, proximal methods, FISTA, ADMM, HQS,...)

Model-based Restoration

- Classical priors: Tikhonov, Lasso, TV
- Image Denoisers / Data-driven priors:
  - RED (Romano2017, ...), Plug-n-Play (Venkatakrishnan2013, ...)
  - GAN/VAE (Bora2017, Gonzalez2022, ...)
  - Diffusion (Kawar2022, Kadkhodaie2021,...)
- Efficient Model-based methods - Polyblur (Delbracio2021)

Romano, Y., Elad, M. and Milanfar, P. The little engine that could: Regularization by denoising (RED). SIAM IS 2017
Venkatakrishnan, S.V., Bouman, C.A. and Wohlberg, B. Plug-and-play priors for model based reconstruction. IEEE Global SIP 2013
Bora, A., Jalal, A., Price, E. and Dimakis, A.G, Compressed sensing using generative models. In International Conference on Machine Learning ICML 2017
González, M., Almansa, A. and Tan, P., Solving inverse problems by joint posterior maximization with autoencoding prior. SIIMS 2022
Kawar, B., Elad, M., Ermon, S. and Song, J., Denoising diffusion restoration models. NeurIPS 2022
Kadkhodaie, Z. and Simoncelli, E., Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. NeurIPS 2021
Delbracio, M., Garcia-Dorado, I., Choi, S., Kelly, D. and Milanfar, P. Polyblur: Removing mild blur by polynomial reblurring. IEEE TCI 2021

Model-based Restoration

- Classical priors: Tikhonov, Lasso, TV
- Image Denoisers / Data-driven priors:
  - RED (Romano2017, ...), Plug-n-Play (Venkatakrish
  - GAN/VAE (Bora2017, Gonzalez2022, ...)
  - Diffusion (Kawar2022, Kadkhodaie2021,...)
- Efficient Model-based methods - Polyblur (Delbracio2021)

Very active area of research!

Romano, Y., Elad, M. and Milanfar, P. The little engine that could: Regularization by denoising (RED). SIAM IS 2017
Venkatakrishnan, S.V., Bouman, C.A. and Wohlberg, B. Plug-and-play priors for model based reconstruction. IEEE Global SIP 2013
Bora, A., Jalal, A., Price, E. and Dimakis, A.G, Compressed sensing using generative models. In International Conference on Machine Learning ICML 2017
González, M., Almansa, A. and Tan, P., Solving inverse problems by joint posterior maximization with autoencoding prior. SIIMS 2022
Kawar, B., Elad, M., Ermon, S. and Song, J., Denoising diffusion restoration models. NeurIPS 2022
Kadkhodaie, Z. and Simoncelli, E., Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. NeurIPS 2021
Delbracio, M., Garcia-Dorado, I., Choi, S., Kelly, D. and Milanfar, P. Polyblur: Removing mild blur by polynomial reblurring. IEEE TCI 2021

Removing mild blur in images

Our goals

Given *any* input image, the method should:

1. Remove **mild** blur in the presence of noise.
2. Not introduce artifacts (ringing or halos).
3. Be efficient (mobile phones).

One Solution: Blur Estimation

Assumption: The blur is **small** and **constant** all over the image

$$v = Ku + n$$

v is the input image, *u* is the sharp image, *K* is the convolution with the blur kernel k , and n is additive noise

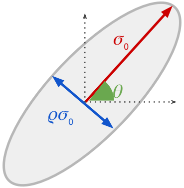
- **Reasonable** for mild camera shake, mild lens aberration, mild out-of-focus
- **Limitation**: might be unrealistic (object motion / strong camera shake, depth-of-field)

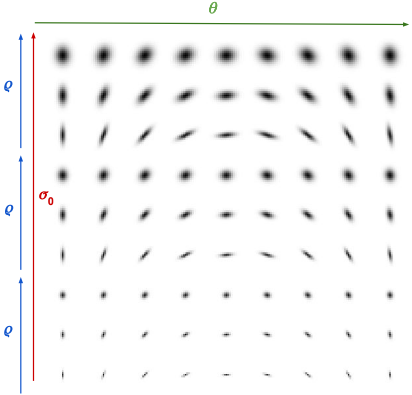
Delbracio, M., Garcia-Dorado, I., Choi, S., Kelly, D. and Milanfar, P., Polyblur: Removing mild blur by polynomial reblurring. *TCI 2021*

Blur estimation

Instead of estimating a full blur kernel **k**, we parametrize the space of possible blurs.

- 2D Gaussian Blur - 3 parameters: $\boldsymbol{\sigma,\varrho,\theta}$
- Fine model if blur is small

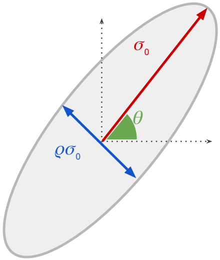




Delbracio, M., Garcia-Dorado, I., Choi, S., Kelly, D. and Milanfar, P., Polyblur: Removing mild blur by polynomial reblurring. *TCI 2021*

Blur estimation

Estimate blur parameters using image gradient:



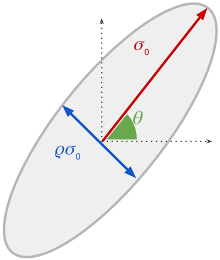
Assumption 1. Maximum sharp image gradient value in any direction is mostly constant and roughly independent of the image.

Assumption 2. Gaussian blur affects the maximum image gradient value inversely proportional to the blur level.

Blur estimation

Estimate blur parameters using image gradient:

1. Compute **maximum of image derivative** at N different directions uniformly sampled in [0,π), i.e., $f_{\theta_1}, f_{\theta_2}, \ldots, f_{\theta_N}$
2. Find f_θ the **minimum** value, and the respective angle θ
   → This is the principal **direction of the blur**
3. We approximately have $\sigma_0 = \sqrt{\frac{a^2}{f_\theta^2} - b^2}$ $\sigma_1 = \sqrt{\frac{a^2}{f_{\theta+\frac{\pi}{2}}^2} - b^2}$
   where (a,b) are calibrated using simulations.
4. The 2D Gaussian parameters are: $\sigma_0, \sigma_1/\sigma_0, \theta$

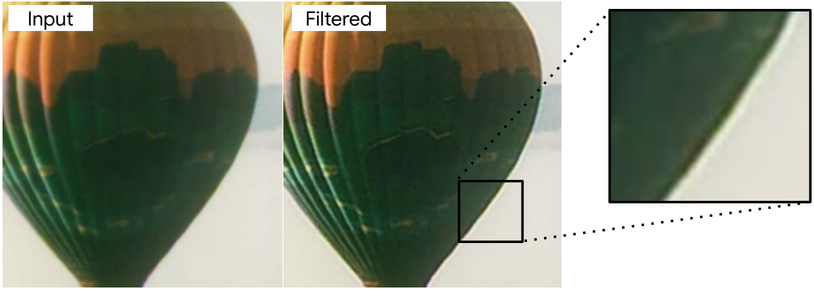


Not perfect...
...but fine estimation.

Blur Estimation Results

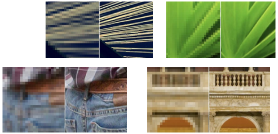
Halo detection and removal

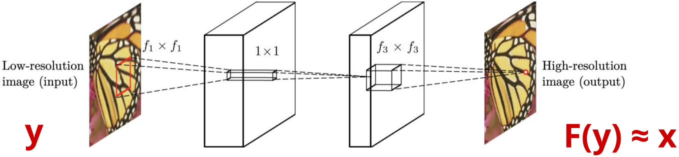
- Halos can be generated due to mis-estimation or due to model mismatch
- Blurry image (v) and restored image (u) have opposite gradients
- Final image blending to avoid gradient reversal:



Deep Learning *Regression* Formulation

From **thousands** of examples **(x, y) learn** the deep model parameters by **minimizing** some **loss** function.



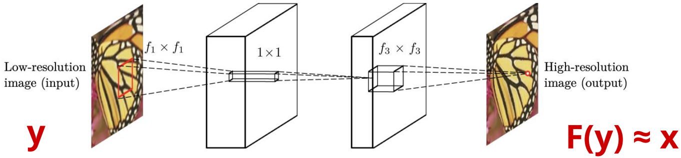


$\boldsymbol{\theta}$ – deep model parameters

Deep Learning *Regression* Formulation

From **thousands** of examples **(x, y)** **learn** the deep model parameters by **minimizing** some **loss** function.

Example, L2-Loss function: $\mathcal{L}(\theta) = \sum_i ||F_\theta(\mathbf{y}_i) - \mathbf{x}_i||^2$ Reconstruction Pixel error



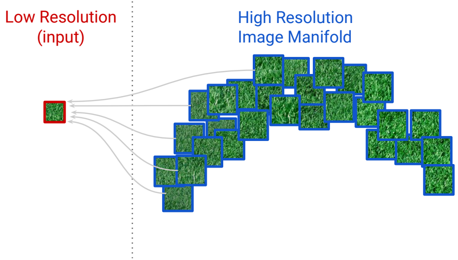
$\boldsymbol{\theta}$ – deep model parameters

Deep Learning *Regression* Formulation

From **thousands** of examples **(x, y)** **learn** the deep model parameters by **minimizing** some **loss** function.

Example, L2-Loss function: $\mathcal{L}(\theta) = \sum_i ||F_\theta(\mathbf{y}_i) - \mathbf{x}_i||^2$

Reconstruction Pixel error



Ill-posed inverse problem maps **many-to-one**.
Many HR images, have the same LR observation.

Deep Learning *Regression* Formulation

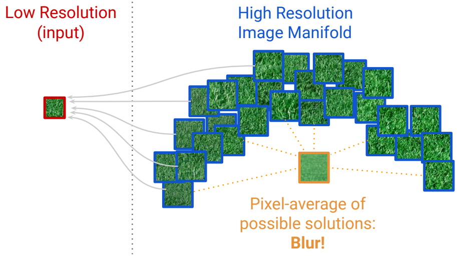
From **thousands** of examples **(x, y) learn** the deep model parameters by **minimizing** some **loss** function.

Example, L2-Loss function: $\mathcal{L}(\theta) = \sum_i ||F_\theta(\mathbf{y}_i) - \mathbf{x}_i||^2$

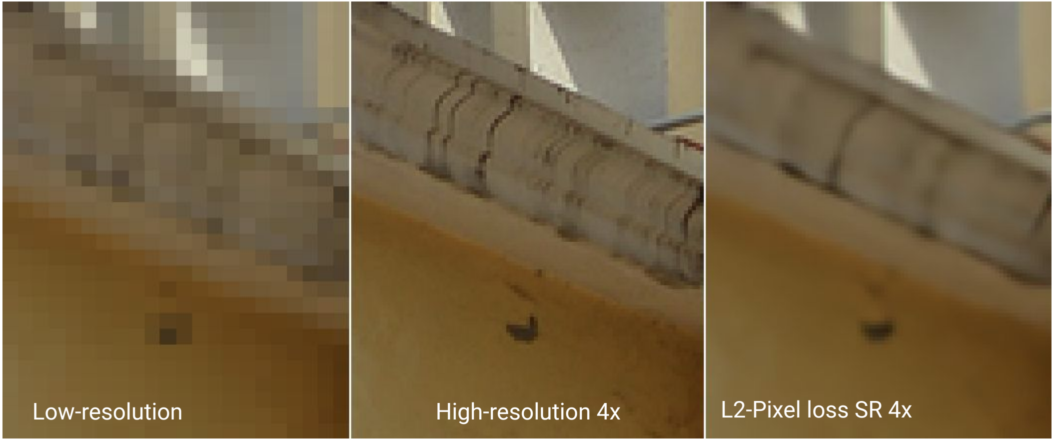
Reconstruction Pixel error

Ill-posed inverse problem maps **many-to-one**. Many HR images, have the same LR observation.

Regression to the mean: Predict the average of the possible candidates → Blur.



Regression to the mean - L2 Pixel loss SRx4 Example

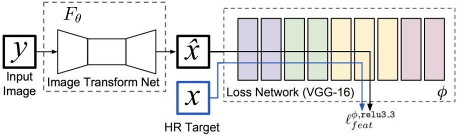


Beyond Pixel Loss

$\Phi(\cdot) \triangleq$ VGG Feature Space

Perceptual/Content Loss [Johnson et al. 2016]

$$\mathcal{L}_{\text{perceptual}} = \sum_i \|\Phi(F_\theta(\mathbf{y}_i)) - \Phi(\mathbf{x}_i)\|^2$$



Pre-trained
Feature Extractor
VGG network

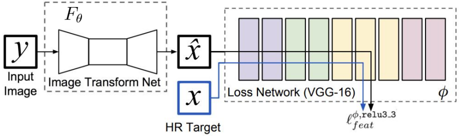
Adversarial Loss [Ledig et al. 2017]

Push *output* to natural image manifold
Train two networks, a generator and a discriminator/critic

Beyond Pixel Loss

Perceptual/Content Loss [Johnson et al. 2016]

$$\mathcal{L}_{\text{perceptual}} = \sum_i \|\Phi(F_\theta(\mathbf{y}_i)) - \Phi(\mathbf{x}_i)\|^2$$



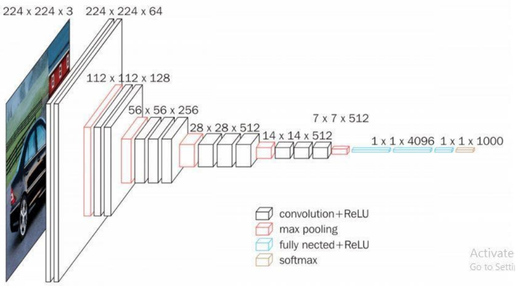
Better than pixel loss, but *regression to the mean* in the deep feature space.

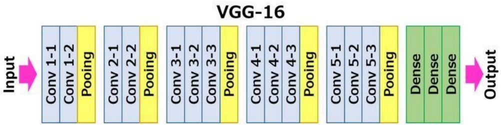
Adversarial Loss [Ledig et al. 2017]

Push *output* to natural image manifold
Train two networks, a generator and a discriminator/critic

VGG 16

- 13 Convolutional Layers
- 5 Max Pooling layers
- 3 Dense layers
- 138 million parameters





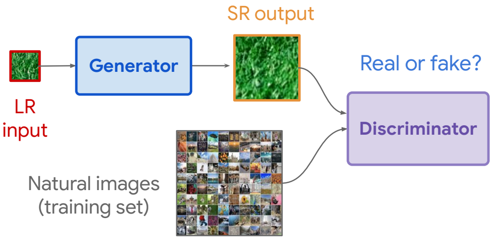
Beyond Pixel Loss

Perceptual/Content Loss [Johnson et al. 2016]

$$\mathcal{L}_{\text{perceptual}} = \sum_i \|\Phi(F_\theta(\mathbf{y}_i)) - \Phi(\mathbf{x}_i)\|^2$$

Better than pixel loss, but *regression to the mean* in the deep feature space.

Adversarial Loss [Ledig et al. 2017]



Push *output* to natural image manifold
Train two networks, a generator and a discriminator/critic

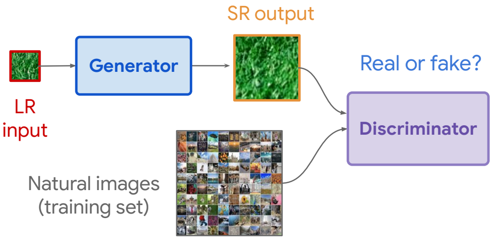
Beyond Pixel Loss

Perceptual/Content Loss [Johnson et al. 2016]

$$\mathcal{L}_{\text{perceptual}} = \sum_i \|\Phi(F_\theta(\mathbf{y}_i)) - \Phi(\mathbf{x}_i)\|^2$$

Better than pixel loss, but *regression to the mean* in the deep feature space.

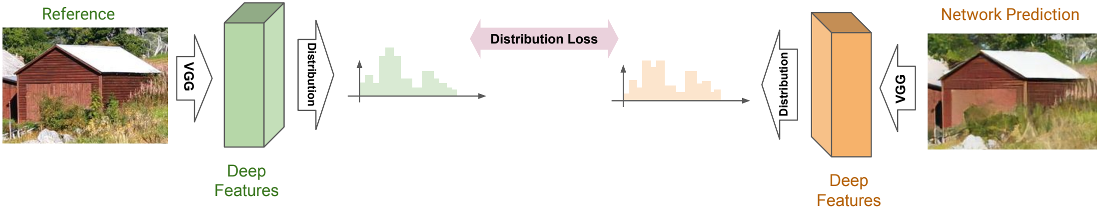
Adversarial Loss [Ledig et al. 2017]



Hard to control image **hallucinations**
Hard/Unstable to train (mode collapse, min-max optimization)

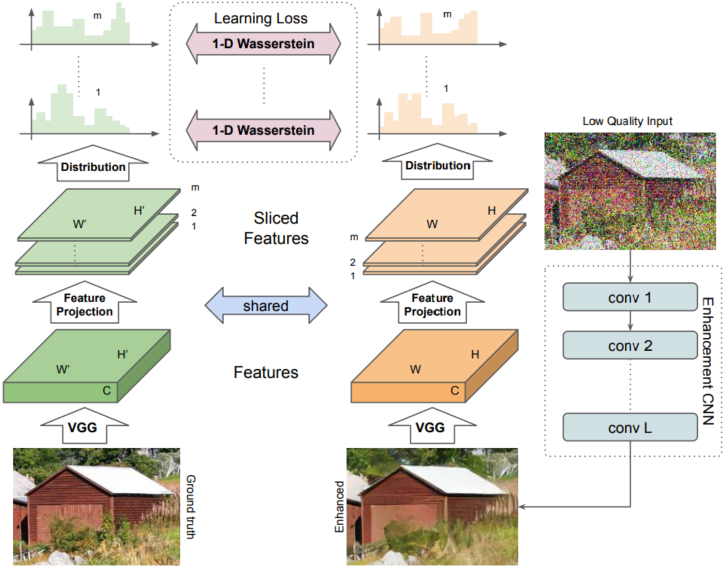
Projected Distribution Loss (PDL) - 1-D Wasserstein Distance

- Encodes **relevant** high-frequency information (e.g., sharpness, grain),
- **Invariant** to small shifts, rotations, noise, other small changes in images,
- Serves as a **complement** to regular pixel loss (L1/L2).



Delbracio, M., Talebi, H. and Milanfar, P. Projected distribution loss for image enhancement. *ICCP 2021*

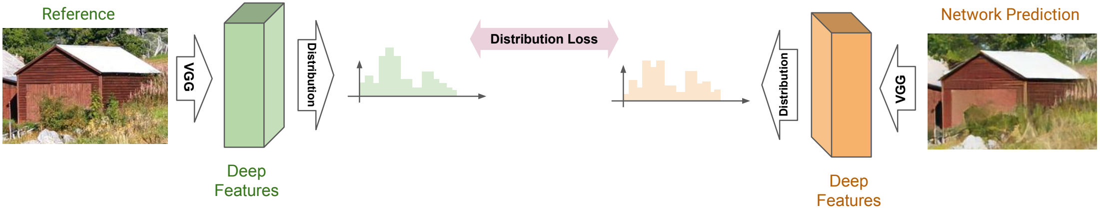
Projected Distribution Loss (PDL) - 1-D Wasserstein Distance



Projected Distribution Loss (PDL)

Distance between ***distribution of features*** (deep features) extracted from generated and target images.

- Encodes **relevant** high-frequency information (e.g., sharpness, grain),
- **Invariant** to small shifts, rotations, noise, other small changes in images,
- Serves as a **complement** to regular pixel loss (L1/L2).

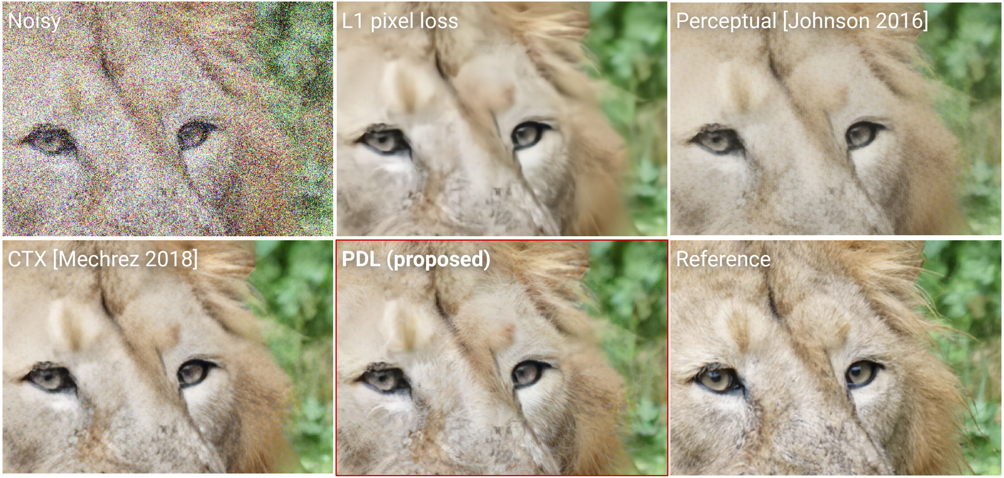


Delbracio, M., Talebi, H. and Milanfar, P. Projected distribution loss for image enhancement. *ICCP 2021*

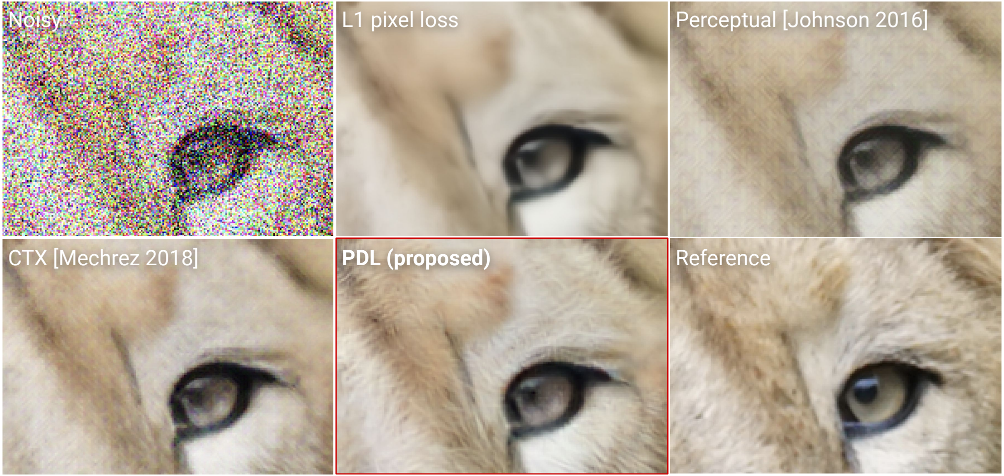

Blurry


Deblurred PDL

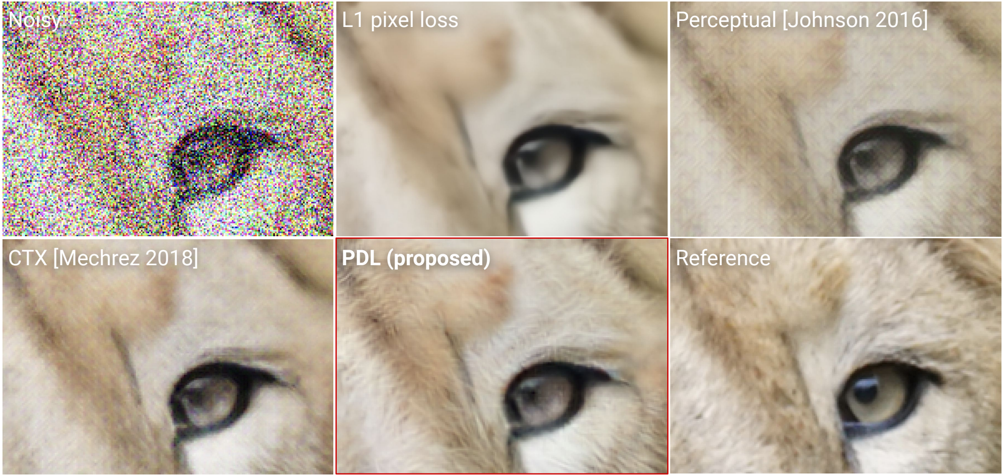
PDL Results - Denoising



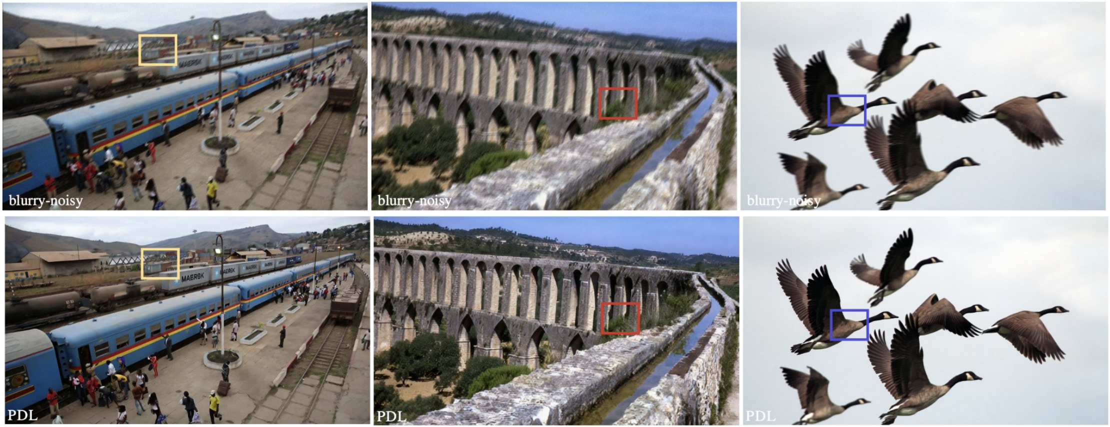
PDL Results - Denoising



PDL Results - Denoising

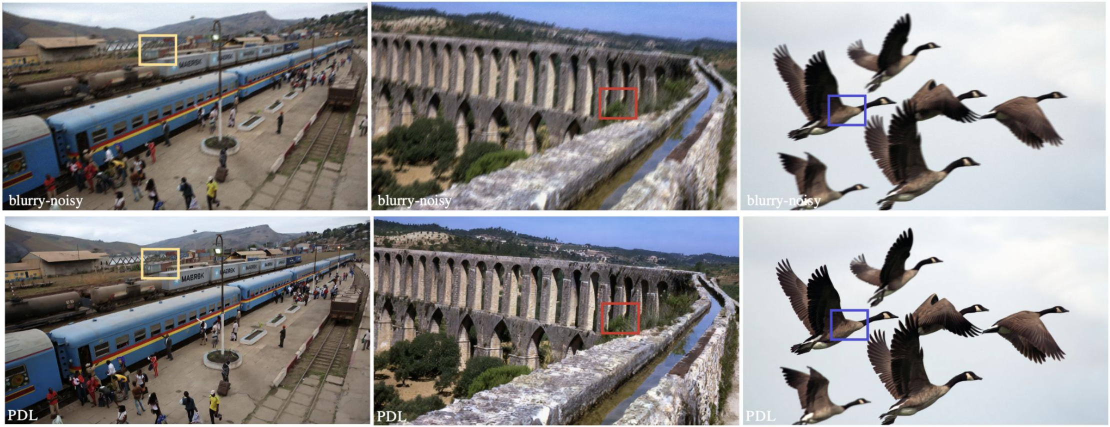


PDL Results - Deblurring



Delbracio, M., Talebi, H. and Milanfar, P. Projected distribution loss for image enhancement. *ICCP 2021*

PDL Results - Deblurring



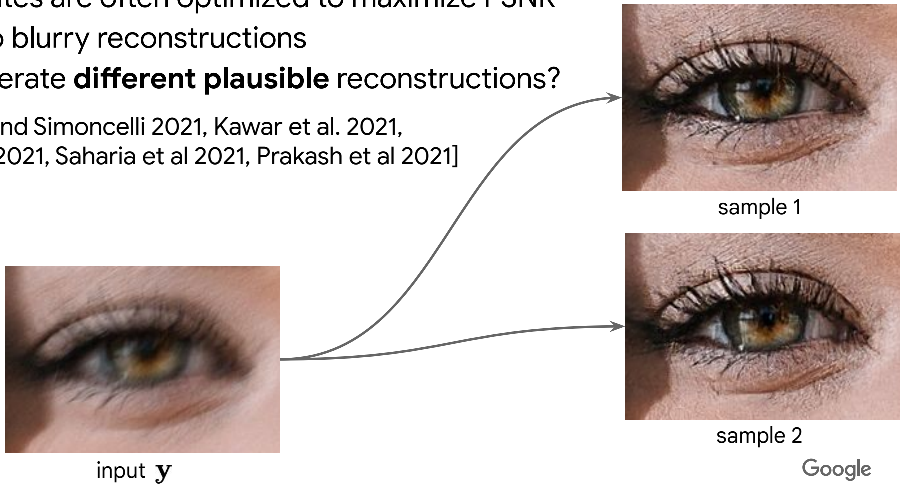
Delbracio, M., Talebi, H. and Milanfar, P. Projected distribution loss for image enhancement. *ICCP 2021*

Denoising Diffusion

Point vs. Sampling restorations

- Point estimates are often optimized to maximize PSNR
- This leads to blurry reconstructions
- Can we generate **different plausible** reconstructions?

[Kadkhodaie and Simoncelli 2021, Kawar et al. 2021,
Ohayon et al. 2021, Saharia et al 2021, Prakash et al 2021]



Google

Point vs. Sampling restorations

- Point estimates are often optimized to maximize PSNR
- This leads to blurry reconstructions
- Can we generate **different plausible** reconstructions?

Conditional diffusion model

- Sample from the distribution of plausible reconstructions (i.e., the *posterior*)

sample 1

sample 2

input $\mathbf{y}$

$$\hat{\mathbf{x}} \sim p(\mathbf{x}|\mathbf{y})$$

Google

Diffusion Models

https://www.flickr.com/photos/jikatu/27381926452

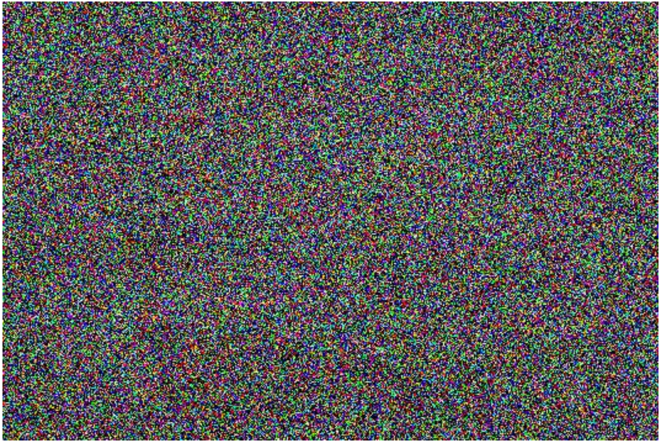
Diffusion Models

https://www.flickr.com/photos/jikatu/27381926452

Diffusion Models

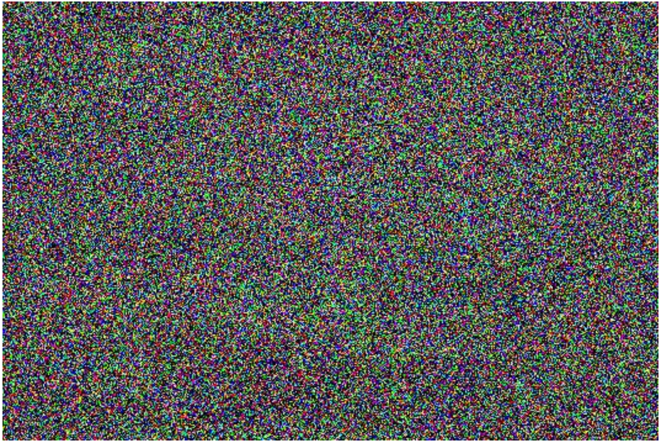
https://www.flickr.com/photos/jikatu/27381926452

Diffusion Models



https://www.flickr.com/photos/jikatu/27381926452

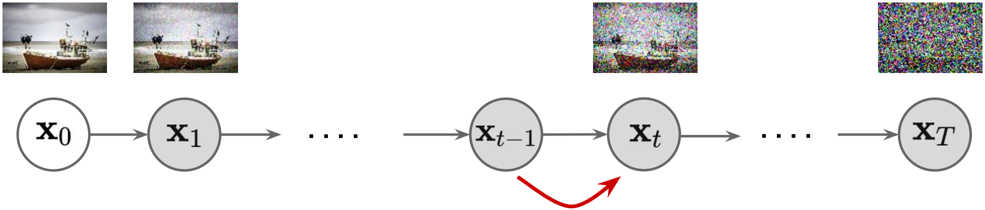
Diffusion Models: Reverse



https://www.flickr.com/photos/jikatu/27381926452

What is a Diffusion Probabilistic Model (DPM)?

- Given a real data point **add Gaussian noise** to the sample in T steps



Forward Process

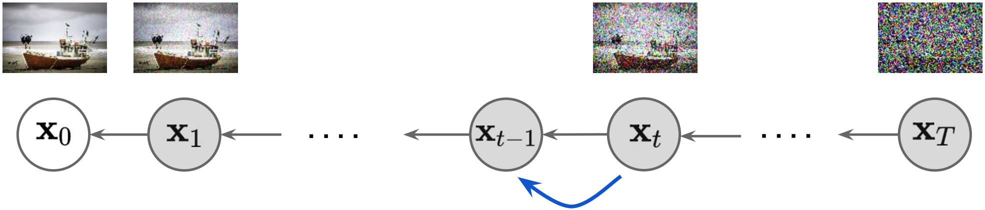
$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$$

Add noise

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. and Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics. In ICML 2015.
Ho, J., Jain, A. and Abbeel, P., Denoising diffusion probabilistic models. In NeurIPS 2020.

What is a Diffusion Probabilistic Model (DPM)?

- Iteratively transforms Gaussian noise to a realistic sample by **denoising**

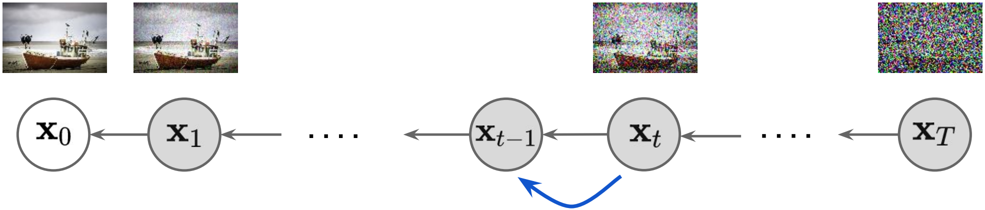


Reverse Process

$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ Unknown

What is a Diffusion Probabilistic Model (DPM)?

- Iteratively transforms Gaussian noise to a realistic sample by **denoising**



Reverse Process

$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ Unknown

$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$ Denoiser Neural Network

DPM Training

- **Pseudocode**
  - Sample $\mathbf{x}_0$ and time step $t \sim \{1, \ldots, T\}$
    - Sample $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and compute $\mathbf{x}_t$ from $\mathbf{x}_0$ and ϵ $\mathbf{x}_0 + \alpha_t \epsilon$
      - Gradient step to minimize
  - Model tries to "undo" the added noise (**denoising**) $\left\| \epsilon - \epsilon_\theta(\mathbf{x}_t, t) \right\|$

Ho, J., Jain, A. and Abbeel, P., Denoising diffusion probabilistic models. In NeurIPS 2020.

DPM Sampling

- Pseudocode
  - Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$

from [Ho et al., 2020]

DPM Sampling

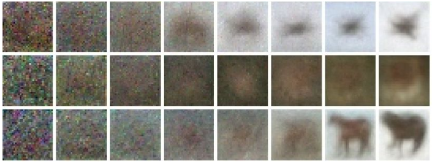
- Pseudocode
  - Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  - For $t = T, \ldots, 1$
    - Given $\mathbf{x}_t$, compute $\hat{\mathbf{x}}_0$ the current estimate of $\mathbf{x}_0$
    - Sample $\mathbf{x}_{t-1}$ from $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \hat{\mathbf{x}}_0)$

from [Ho et al., 2020]

DPM Sampling

- Pseudocode
  - Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  - For $t = T, \ldots, 1$
    - Given $\mathbf{x}_t$, compute $\hat{\mathbf{x}}_0$ the current estimate of $\mathbf{x}_0$
    - Sample $\mathbf{x}_{t-1}$ from $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \hat{\mathbf{x}}_0)$

from [Ho et al., 2020]

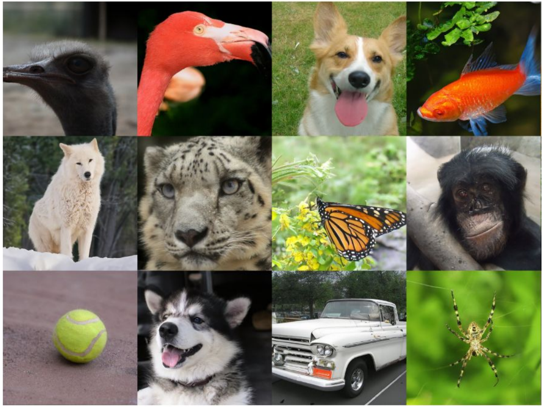


DPM Sampling

- Pseudocode
  - Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  - For $t = T, \ldots, 1$
    - Given $\mathbf{x}_t$, compute $\hat{\mathbf{x}}_0$ the current estimate of $\mathbf{x}_0$
    - Sample $\mathbf{x}_{t-1}$ from $q(\mathbf{x}_{t-1} | \mathbf{x}_t, \hat{\mathbf{x}}_0)$

from [Ho et al., 2020]

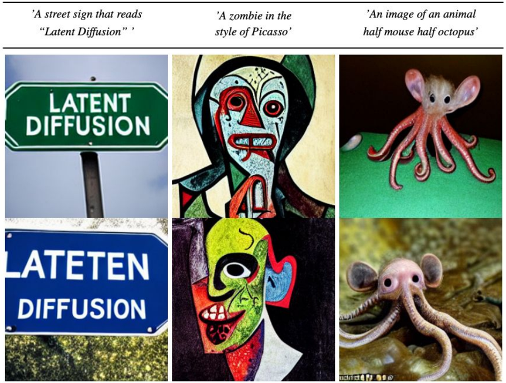
Denoising Diffusion - Image Generation - Recent Work



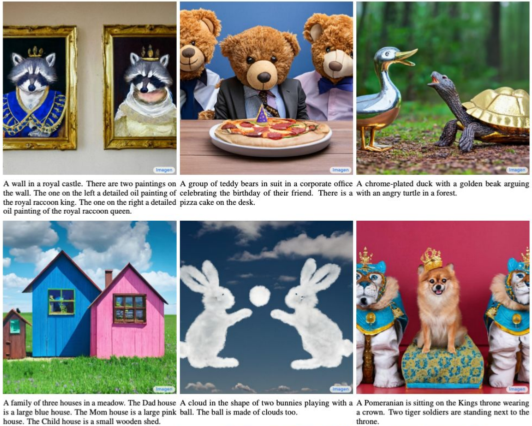
Dhariwal and Nichol, 2021

Saharia et al., 2021

Denoising Diffusion - Image Generation - Recent Work

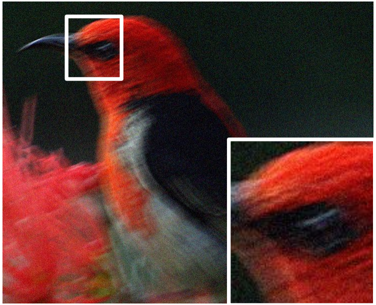


Rombach et al., 2022 (Stable Diffusion)



Saharia et al., 2022 (Imagen)

Some More Results



Input

Initial Prediction

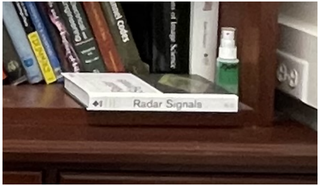
Sample

Static vs Moving Camera - Dr. Arce's Office

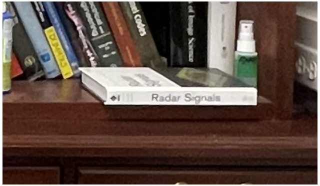
Static Camera

Moving Camera

Static Camera



Moving Camera



- Slides adapted from "A Walk Through Image Deblurring: From Model-Based to Generative Restoration" by Mauricio Delbracio with Google Research.