Denoising Capacity of Implicit Image Representation

Zhengyang (Leo) Dong[1]

[1]*Department of Computer Science, Stanford University*

Neural Implicit Functions for Denoising

- Goal: Explore the extent to which neural implicit representation of images can be used for denoising.
- Motivation: Implicit functions parameterized by neural networks have shown promising results in fitting image signals. We want to explore the capacity in which this representation can help image denoising.
- Previous work: *SIREN*, *DnCNN*, *BM3D*
- SIREN is a type of neural implicit function that uses periodic activation functions. This is the primary model we study.
- DnCNN is a data-driven denoiser using CNN.
- BM3D is the state-of-the-art non-data-driven denoiser.
- Our work: We use a non-data-driven approach and do not assume access to an external dataset of natural images. We use SIREN to 1) directly fit a noisy image, 2) fit a noisy image with TV regularization, 3) fit a noisy with TV and spline positional encoding.

SIREN Formulation

$$\Phi(\mathbf{x}) = \mathbf{W}_n(\phi_{n-1} \circ \phi_{n-2} \circ \ldots \circ \phi_0)(\mathbf{x}) + \mathbf{b}_n,$$

$$\mathbf{x}_i \mapsto \phi_i(\mathbf{x}_i) = \sin(\mathbf{W}_i\mathbf{x}_i + \mathbf{b}_i).$$

Each ϕ_i is a layer of neural network comprised with a sine activation function. The input x is a grid of 2D coordinates (because we are fitting images), and the output $\Phi(\boldsymbol{x})$ is a grid of 3D values representing the RGB channels.

Note that $\Phi(\boldsymbol{x})$ is now a continuous and differentiable representation of the input image, which we can optimize for fitting the noisy image directly or further regularize the gradient of $\Phi(\boldsymbol{x})$ to minimize total variation (TV).

SPE Formulation

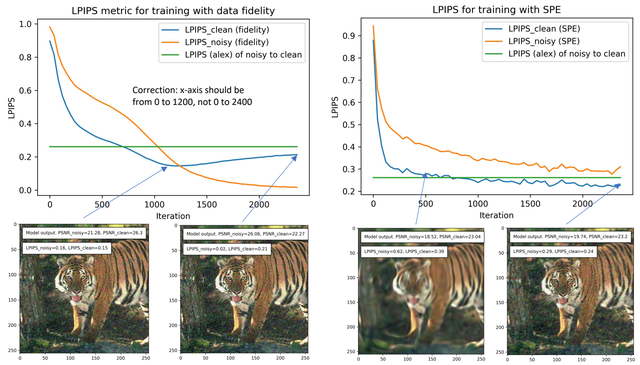
We also experiment with modifying a SIREN network with an additional spline positional encoder (SPE) to its input coordinate grid $\boldsymbol{x}$. For each 2D coordinate, We first select m random unit directions, and project each 2D coordinate x to a 1D point $x_i \coloneqq \langle x, D_i \rangle$ for m times. Then the SPE of x is:

$$S(\mathbf{x}) = [\psi_1(x_1), ..., \psi_m(x_m)]$$

Each ψ_i is a parametric spline function. We simply transform the input coordinate $\boldsymbol{x}$ by $S(\boldsymbol{x})$ and then pass it as input coordinate to SIREN, $\Phi(S(\boldsymbol{x}))$.

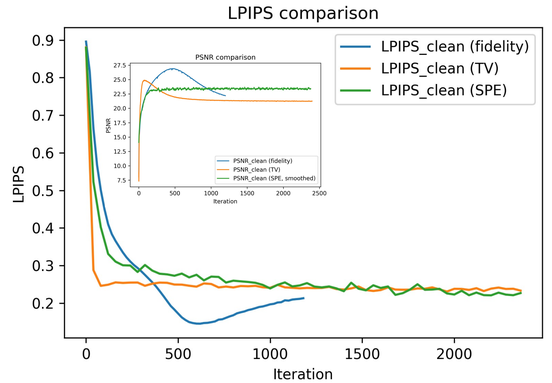
SIREN Training Has Two Phases

During training, we plot `LPIPS_clean` (LPIPS compared to clean image) and `LPIPS_noisy` (LPIPS compared to noisy image). We observe that while `LPIPS_noisy` keeps decreasing, `LPIPS_clean` would eventually increase, which is the phase where SIREN is overfitting to noise. This is not observed when training with SPE, although SPE is much slower to converge (2400 steps instead of 1200 steps).



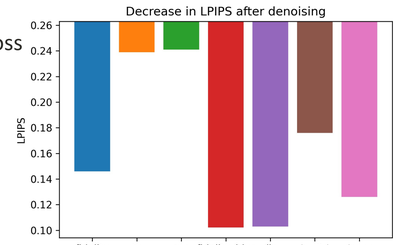
SPE is a Form of Regularization

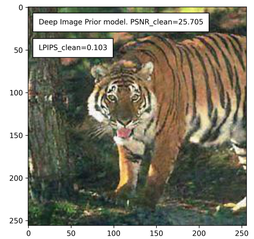
Here we compare the LPIPS and PSNR curves of SIREN trained under these three conditions. We observe that SPE implicitly acts as a kind of regularization that prevents both LPIPS and PSNR to clean from getting too bad. But in the best-case early-stopping, SIREN with just the data fidelity loss still achieves the best metric. However, we like to note the best metric does not always reflect most pleasing visual results (even in the case of LPIPS).

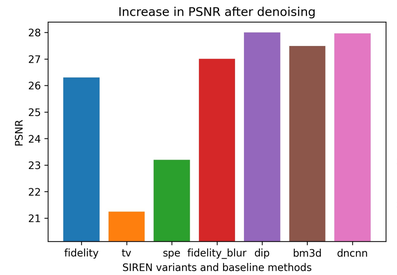


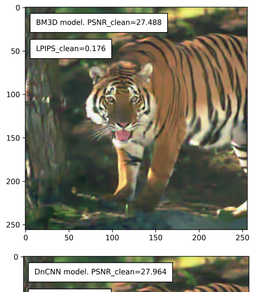
Comparison with Baselines

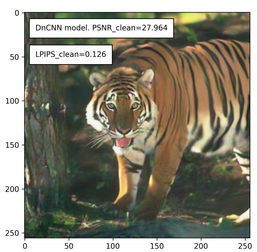
We show results of 3 baseline methods. In addition to BM3D and DnCNN, Deep Image Prior (DIP) is an early-stopped CNN trained to fit a single image.





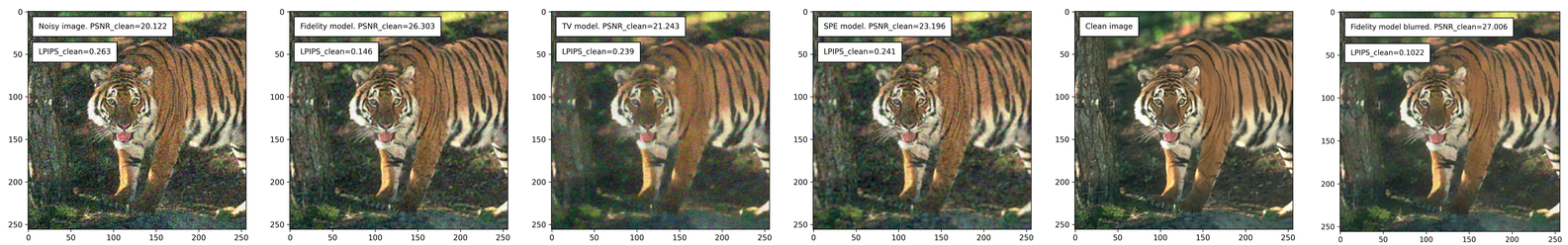






*It is important to note that DnCNN is the only data-driven method, so it has access to more information and trains for much longer.

Early-Stopped Denoising Results



Methodology and Experiments

We are given a corrupted image $\hat{I}$ from $I + \epsilon$, where I is the clean image. For simplicity, we assume $\hat{I}$ is defined on the grid $[-1, 1]^2$ but we only observe a discrete set of values $\hat{I}(G)$, where G is a set of evenly sampled points.

1. Fitting SIREN directly to noisy image: $\mathcal{L}_{\text{fidelity}} = \int \|\Phi(\mathbf{x}) - \hat{I}(\mathbf{x})\|_2 d\mathbf{x}$
2. Fitting SIREN to noisy image with total variation (TV) regularization: $\mathcal{L}_{\text{tv}} = \int \|\Phi(\mathbf{x}) - \hat{I}(\mathbf{x})\|_2 + \kappa\|\nabla_{\mathbf{x}}\Phi(\mathbf{x})\|_1 d\mathbf{x}$
3. Fitting SIREN to spline coordinate encoded grid: $\mathcal{L}_{\text{spe}} = \int \|\Phi(S(\mathbf{x})) - \hat{I}(\mathbf{x})\|_2 d\mathbf{x}$

We evaluate the results quantitatively with PSNR (higher better) and LPIPS (lower better) to the clean image. Compared to PSNR, LPIPS focuses more on evaluating how perceptually similar two images are.

Continuous Bilateral Filtering

Because we have learned a continuous functional representation Φ of an image, we can apply a continuous analog of bilateral filtering on Φ. Specifically, we randomly perturb the input grid G for k times, and for each perturbed $\hat{G}_i$, we Φ and weight the result $\Phi(\hat{G}_i)$ by a location weight $W_l(G, \hat{G}_i)$ and intensity weight $W_p\big(\Phi(G), \Phi(\hat{G}_i)\big)$. The filtered result is thus $\sum_{i=1}^{k}(W_l \cdot W_p)\Phi(\hat{G}_i)$. We apply this on the fidelity model's result and include it in the comparisons.

Note on metric: we found that neither PSNR nor LPIPS truthfully reflect human preference. For example, while DIP achieves highest LPIPS, we feel that DnCNN produces a more visually appealing image.