Korea
university
Biomedical Engineering

RADIOLOGY

CCIDS 의료영상데이터사이언스센터 Center for Clinical Imaging Data Science

2020. 03. 29

Noise2Noise: Learning Image Restoration without Clean Data

Sang Wook Kim
Korea university
Department of Biomedical Engineering

Jaakko Lehtinen, et al., “Noise2Noise: Learning Image Restoration without Clean Data”, ICML, 2018 contents

01
Introduction

02
Theoretical Background

03
Practical Experiments

04
Discussion

Introduction

1. Apply basic statistical reasoning to signal reconstruction by machine learning

   => Learning to map corrupted observations to clean signals

2. Possible to learn to restore images by only looking at corrupted examples

3. W/O image priors or likelihood models of the corruption

4. Single model learns photographic noise removal
   - Denoising synthetic Monte Carlo images
   - Reconstruction of under sampled MRI scans

* Corrupted = Noisy = Not Clean

Theoretical Background

$$\underset{z}{\operatorname{argmin}}\, \mathbb{E}_y\{L(\boxed{z,}\; y)\}.$$

$$\underset{\theta}{\operatorname{argmin}}\, \mathbb{E}_{(x,y)}\{L(\boxed{f_\theta(x)},\; y)\}.$$

$$\underset{\theta}{\operatorname{argmin}}\, \boxed{\mathbb{E}_x\{\mathbb{E}_{y|x}}\{L(f_\theta(x),\; y)\}\}.$$

Indeed, if we remove the dependency on input data, and use a trivial f_θ that merely outputs a learned scalar, the task reduces to (2). Conversely, the full training task decomposes to the same minimization problem at every training sample; simple manipulations show that (4) is equivalent to

The network can, in theory, minimize this loss by solving the point estimation problem separately for each input sample.

Theoretical Background

1. Instead of 1:1 mapping between inputs & targets => In reality, the mapping is multiple valued

2. Example : Super Resolution
    - Low-res image x can be explained by many different high-res images
    - Knowledge about the exact positions and orientations of the edges are lost in decimation
    - When trained with L2 Loss ~ the network learns to output the average of all plausible explanations

Theoretical Background

1. Unexpected benefits of averaging plausible outputs
    - The estimate remains unchanged if we replace the targets with random numbers whose expectation match the targets.
    - Below Equation holds, No matter what particular distribution the Y are drawn from

$$z = \mathbb{E}_y\{y\}.$$

2. The optimal network parameter (theta) also remain unchanged
    - Even if input-conditioned target distribution p(y|x) is replaced with arbitrary distribution that has the same expectation.

3. This implies that in principle, corrupting the training targets of Neural Network with zero-mean noise without changing the network learns

Theoretical Background

1. Both the inputs and the targets are now drawn from a corrupted distribution

$$\operatorname*{argmin}_{\theta} \sum_i L\left(\boxed{f_\theta(\hat{x}_i)}, \boxed{\hat{y}_i}\right), \qquad \mathbb{E}\{\hat{y}_i|\hat{x}_i\} = y_i.$$

2. Variance
   - Average variance of the corruption in the targets divided by the number of training samples

3. None of the above relies on a likelihood model of the corruption nor a density model (prior) for the underlying clean image manifold
   - We do not need p(noisy|clean) or p(clean)

Practical Experiments

Additive Gaussian Noise

1. Noise has zero-mean – L2 loss for training to recover the mean
2. Details
   - Baseline – “RED30” (Mao et al., 2016)

     : a 30-layer hierarchical residual network with 128 feature maps
   - Trained with 256 x 256 images (from 50k images in IMAGENET validation set)
   - Randomize noise standard deviation [0, 50] separately for each training example
   - GOAL : Estimate the magnitude of noise while removing it
3. Dataset
   - BSD 300, SET14, KODAK

Additive Gaussian Noise

4. **Result**
   - Qualitatively Similar -> Discuss the average
   - Noisy target performance similar to that of clean target

5. **RED30 -> Shallower U-Net**
   - Roughly 10x faster to train
   - similar result

Additive Gaussian Noise

6. **Convergence speed**
    - **Training Loss does not decrease during training - input , output has different noise & cannot transform input noise -> output noise**
    - **Loss gradients continue to be quite Large => Affect convergence speed ? => No !**
    - **Gradients are indeed noisy / the weight gradients are clean ~ Gaussian noise is i.i.d in all pixels => weight gradients get averaged over 2^16 (256 * 256)**

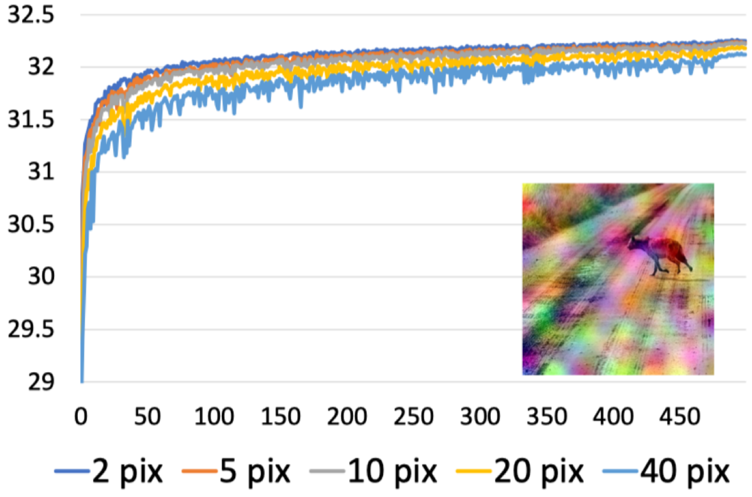
Practical Experiments

Additive Gaussian Noise

6. **Convergence speed**
   - Inter-pixel correlation to the noise
   - Figure 1(b) ~ white Gaussian noise + spatial Gaussian filter of diff. bandwidths and scaling to retain sigma=25
   - As correlation ⬆

     => effective averaging of Weights Gradient ⬇

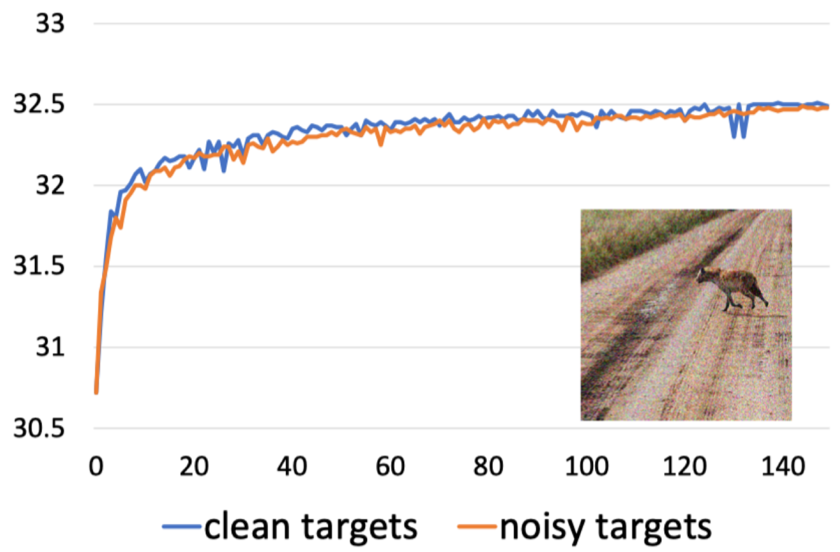
     => convergence slower / extreme blur



(b) Brown Gaussian, $\sigma = 25$

Practical Experiments

Additive Gaussian Noise



(a) White Gaussian, $\sigma = 25$

(a) Gaussian $(\sigma = 25)$

Ground truth Input Our Comparison

Other Synthetic Noise – Poisson Noise

1. Dominant source of noise in photographs
2. Harder to remove because it is signal-dependent
3. L2 loss
4. Result
   - Similar to clean target training

Other Synthetic Noise – Multiplicative Bernoulli noise (binomial noise)

1. To avoid backpropagating gradients for missing pixels

   => Exclude them from the loss

Practical Experiments

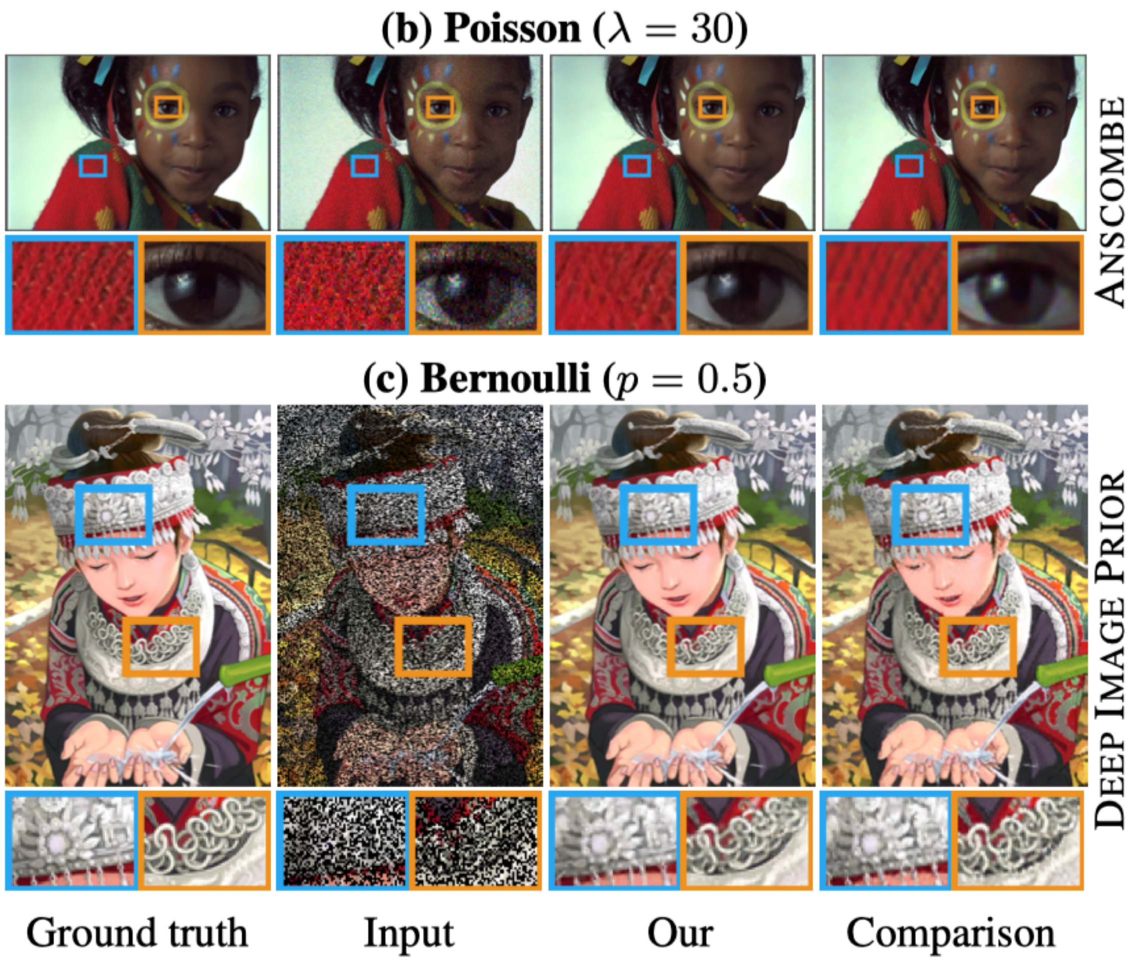


Table 1. PSNR results from three test datasets KODAK, BSD300, and SET14 for Gaussian, Poisson, and Bernoulli noise. The comparison methods are BM3D, Inverse Anscombe transform (ANSC), and deep image prior (DIP).

	Gaussian (σ=25)			Poisson (λ=30)			Bernoulli (p=0.5)		
	clean	noisy	BM3D	clean	noisy	ANSC	clean	noisy	DIP
Kodak	32.50	32.48	31.82	31.52	31.50	29.15	33.01	33.17	30.78
BSD300	31.07	31.06	30.34	30.18	30.16	27.56	31.04	31.16	28.97
Set14	31.31	31.28	30.50	30.07	30.06	28.36	31.51	31.72	30.67
Average	**31.63**	**31.61**	**30.89**	**30.59**	**30.57**	**28.36**	**31.85**	**32.02**	**30.14**

Other Synthetic Noise – Text Removal

1. **Blind Text removal**
   - Varying number of random strings in random places + on top of each other
   - font size, color
2. **Detail**
   - Network is trained using independently corrupted input and target pairs
   - Probability of corrupted pixels [0, 0.5] during training, 0.25 during testing
   - L2 loss (test the mean) is not the correct answer

     $\Rightarrow$ the overlaid text has colors unrelated to the actual image

     $\Rightarrow$ resulting image incorrectly tend towards a linear combination of the right answer and the average text color (medium gray)

Other Synthetic Noise – Text Removal

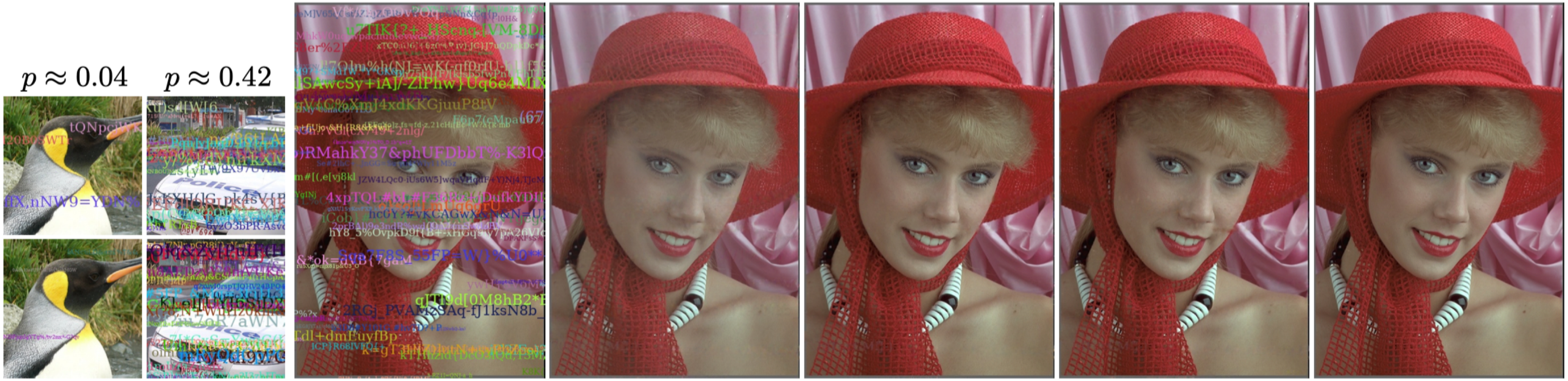


Figure 3. Removing random text overlays corresponds to seeking the median pixel color, accomplished using the L_1 loss. The mean (L_2 loss) is not the correct answer: note shift towards mean text color. Only corrupted images shown during training.

Other Synthetic Noise – Random Valued impulse Noise

1. Replaces some pixels with noise and retains the colors of others
2. Each pixel is replaced w/ a random color drawn from the uniform dist.
3. The pixels' Color dist. : Dirac (@ the original color) + uniform dist.
4. Loss function
   - L1, L2 both does not work (L1 works well when $p < 0.5$)
   - Mode of the distribution (the Dirac spike)
   - L0 loss $(|f_\theta(\hat{x}) - \hat{y}| + \epsilon)^\gamma$

Other Synthetic Noise – Random Valued impulse Noise

Figure 4. For random impulse noise, the approx. mode-seeking L_0 loss performs better than the mean (L_2) or median (L_1) seeking losses.

Practical Experiments

Monte Carlo Rendering

1. Physically accurate renderings of virtual environments
2. Often generated through Monte Carlo path tracing
3. Drawing random sequences of scattering events ("light paths") in the scene
4. Intensity of each pixel is the expectation of the random path sampling process
5. Varies from pixel to pixel, heavily depends on the scene configuration and rendering parameters
6. For those reasons, It is difficult to remove MC noise

Practical Experiments

Monte Carlo Rendering

(a) Input (64 spp), 23.93 dB (b) Noisy targets, 32.42 dB (c) Clean targets, 32.95 dB (d) Reference (131k spp)

Figure 7. Denoising a Monte Carlo rendered image. (a) Image rendered with 64 samples per pixel. (b) Denoised 64 spp input, trained using 64 spp targets. (c) Same as previous, but trained on clean targets. (d) Reference image rendered with 131 072 samples per pixel. PSNR values refer to the images shown here, see text for averages over the entire validation set.

Practical Experiments

Magnetic Resonance Imaging (MRI)

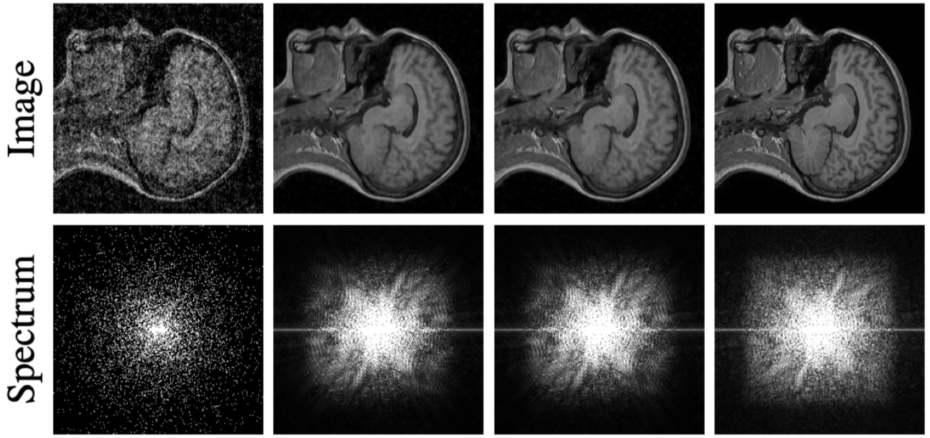
1. MRI produces volumetric images by sampling Fourier transform (k-space) of the signal

2. Loss function : $$(\mathcal{F}^{-1}(R_{\hat{x}}(\mathcal{F}(f_\theta(\hat{x})))) - \hat{y})^2$$

3. Compute loss with raw signal

4. Result

   - Clean target : 31.77 dB
   - Corrupted target : 31.74 dB



(a) Input 18.93 dB (b) Noisy trg. 29.77 dB (c) Clean trg. 29.81 dB (d) Reference

Conclusion

1. It is possible to recover signals under complex corruptions without observing clean signals

2. Performance is same or close to clean target training

3. All based on the same general-purpose deep convolutional model

Thank you