Model-Agnostic Meta-Learning for Image Denoising

Jessica Tawade, Rahul Shiv
Stanford University

Motivation

- Reconstruct image x from noisy observation y affected by noise distribution $\mathbb{N}$ with parameters ϵ :

$$y = \mathbb{N}_{\epsilon}(x)$$

- Deep learning methods exist for denoising images, but such methods require **large amounts of synthetic noisy data** and **do not generalize well** to unseen noise distributions or **real noisy images**
- We propose the use of a **meta-learning algorithm** to learn how to perform few-shot image denoising with various noise distributions
- At test time our model has the ability to **denoise synthetic noisy images of unseen distributions and levels** and also to adapt to denoising of a small set of **real noisy images**

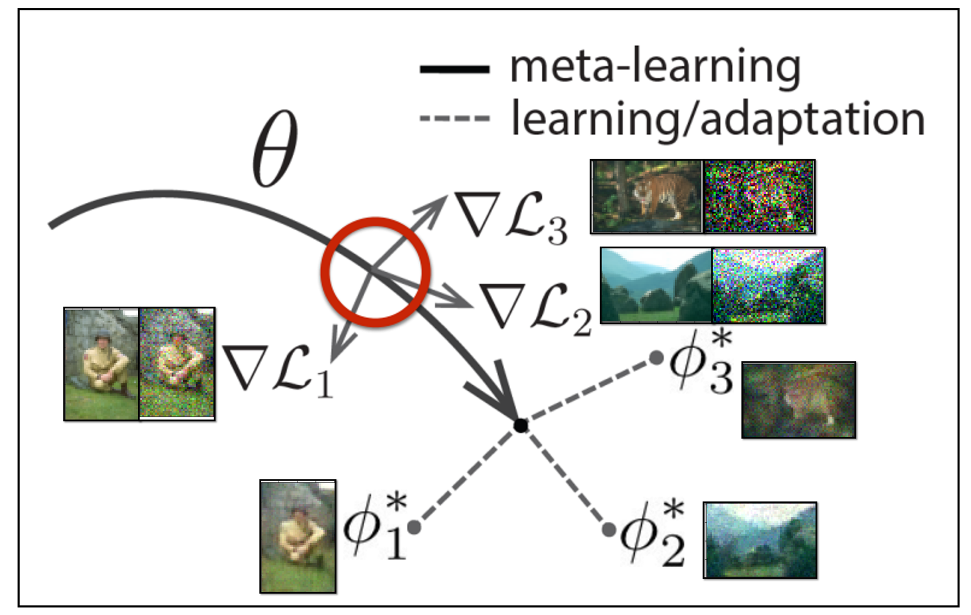
Related Work

- **Non-learning methods**: Bilateral Filtering (BF), Non-Local Means (NLM), and Block-Matching and 3-D Filtering (BM3D)
  - **Pros**: Works independently of noise distribution and level
  - **Cons**: Can cause blurring artifacts, slow run time
- **Learned Methods**: DnCNN-B from Denoising Convolutional Neural Networks [1]
  - **Pros**: DnCNN-B handles Gaussian denoising on unknown levels
  - **Cons**: Needs lot of data to train, only suited for Gaussian noise
- **Model Agnostic Meta-Learning (MAML)** [2]: Finds a set of network initialization parameters that allow the model to adapt to any unseen task quickly with just a few steps of fine-tuning

References

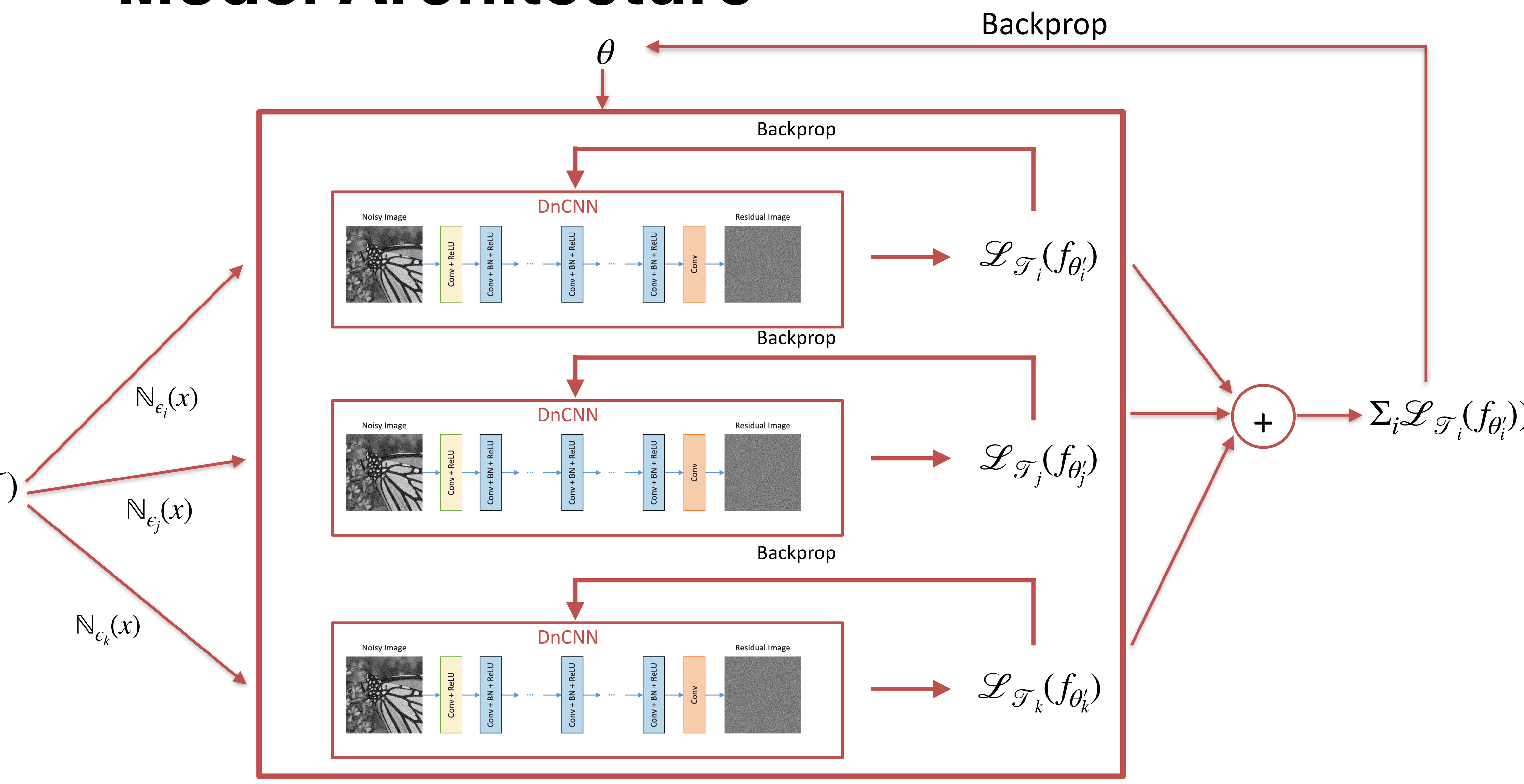
[1] Zhang et. al., Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising, IEEE Transactions on Image Processing, 2016
[2] Finn et. al., Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks, ICML, 2017
[3] Arbelaez et al., Contour Detection and Hierarchical Image Segmentation, IEEE TPAMI, Vol. 33, No. 5, pp. 898-916, May 2011.
[4] Casas et al., "Few-Shot Meta-Denoising". In: CoRR abs/1908.00111, 2019

Model Architecture



Algorithm 1 Model-Agnostic Meta-Learning

Require: $p(\mathcal{T})$: distribution over tasks
Require: α, β: step size hyperparameters
randomly initialize θ
while not done **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$
  for all $\mathcal{T}_i$ **do**
    Evaluate $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$ with respect to K examples
    Compute adapted parameters with gradient descent: $\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(f_\theta)$
  end for
  Update $\theta \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$
end while



Experimental Results

	PSNR				
Noise Parameters	**Noisy**	**BF**	**NLM**	**BM3D**	**MAML, 10-shot**
Gaussian, $\sigma = 0.1$	22.48	29.85	29.30	33.17	30.65
Gaussian, $\sigma = 0.15$	18.84	28.09	29.08	30.89	29.32
Gaussian, $\sigma = 0.3$	13.74	23.45	24.25	25.71	24.60
Poisson, $\beta = 5$	13.55	23.10	24.39	25.26	24.82
Poisson, $\beta = 7$	14.63	24.32	25.47	26.55	26.26
Poisson, $\beta = 15$	17.54	26.97	23.84	29.42	28.81