Deep Gaussian Conditional Random Field Network: A Model-based Deep Network for Discriminative Denoising

Raviteja Vemulapalli
University of Maryland, College Park.

Oncel Tuzel, Ming-Yu Liu
Mitsubishi Electric Research Laboratories, Cambridge.

Main contribution: A Gaussian CRF model-based end-to-end trainable deep network architecture that explicitly models the input noise variance.

Motivation

- Existing discriminative denoising methods such as multilayer perceptrons (MLP) train a separate model for each individual noise level.
- They do not model the input noise variance.
- Having a separate model/network for each noise level is not practical.

- We designed a deep network that
  - explicitly models the input noise variance
  - can handle a range of noise levels
  - is end-to-end trainable using standard gradient-based techniques.

Gaussian CRF Model

- $\boldsymbol{X}$: noisy input image, $\boldsymbol{Y}$: output image
- σ^2: noise variance, $\boldsymbol{G}$: Mean subtraction matrix
- $\boldsymbol{x}_{ij}, \boldsymbol{y}_{ij}$: column vectors representing $d \times d$ patches centered on pixel (i,j) in images $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively.

$$E = \underbrace{\frac{1}{2\sigma^2}\sum_{ij}\left(\boldsymbol{Y}(i,j) - \boldsymbol{X}(i,j)\right)^2}_{\text{Quadratic unary potential}} + \underbrace{\frac{1}{2d^2}\sum_{ij}\boldsymbol{y}_{ij}^T\boldsymbol{G}^T\left(\boldsymbol{\Sigma}_{ij}(\boldsymbol{x}_{ij})\right)^{-1}\boldsymbol{G}\boldsymbol{y}_{ij}}_{\text{Quadratic pairwise potential}}$$

- $\boldsymbol{\Sigma}_{ij}(\boldsymbol{x}_{ij})$: data-dependent parameters of the pairwise potential function

Gaussian CRF Inference

- We use the iterative optimization method of [1].
- $\boldsymbol{z}_{ij}$: auxiliary variable corresponding to $\boldsymbol{y}_{ij}$

$$J(\boldsymbol{Y}, \{\boldsymbol{z}_{ij}\}, \beta) = \sum_{ij}\left\{\begin{array}{c}\frac{d^2}{\sigma^2}\left(\boldsymbol{Y}(i,j) - \boldsymbol{X}(i,j)\right)^2 + \beta\left\|\boldsymbol{z}_{ij} - \boldsymbol{y}_{ij}\right\|_2^2 \\ + \boldsymbol{z}_{ij}^T\boldsymbol{G}^T\left(\boldsymbol{\Sigma}_{ij}(\boldsymbol{x}_{ij})\right)^{-1}\boldsymbol{G}\boldsymbol{z}_{ij}\end{array}\right\}$$

- For a fixed β

  Optimizing J with respect to $\boldsymbol{z}_{ij}$ gives

$$\boldsymbol{z}_{ij} = \left(\boldsymbol{I} - \boldsymbol{G}^T\left(\beta\boldsymbol{\Sigma}_{ij}(\boldsymbol{x}_{ij}) + \boldsymbol{G}\boldsymbol{G}^T\right)^{-1}\boldsymbol{G}\right)\boldsymbol{y}_{ij}$$

(Patch inference (PI) step)

  Optimizing J with respect to $\boldsymbol{Y}$ gives

$$\boldsymbol{Y}(i,j) = \frac{\boldsymbol{X}(i,j)}{1+\beta\sigma^2} + \frac{\beta\sigma^2}{(1+\beta\sigma^2)d^2}\sum_{p,q=-\left\lceil\frac{d-1}{2}\right\rceil}^{\left\lceil\frac{d-1}{2}\right\rceil}\boldsymbol{z}_{pq}(i,j)$$

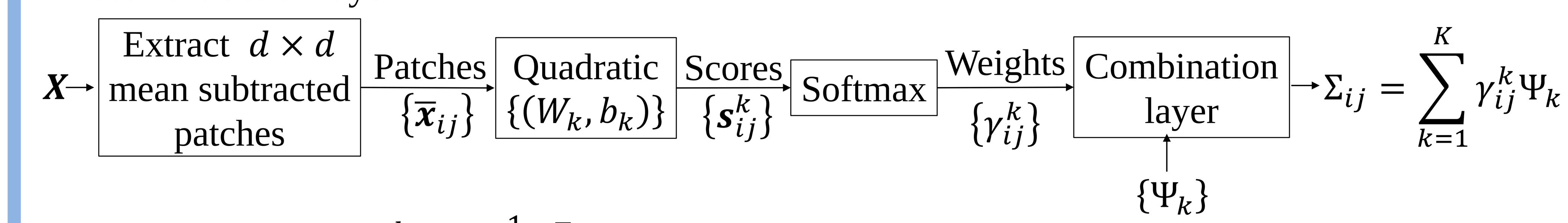
(Image formation (IF) step)

- In [1], the above two optimization steps are repeated while increasing the value of β in each iteration.

Gaussian CRF Network

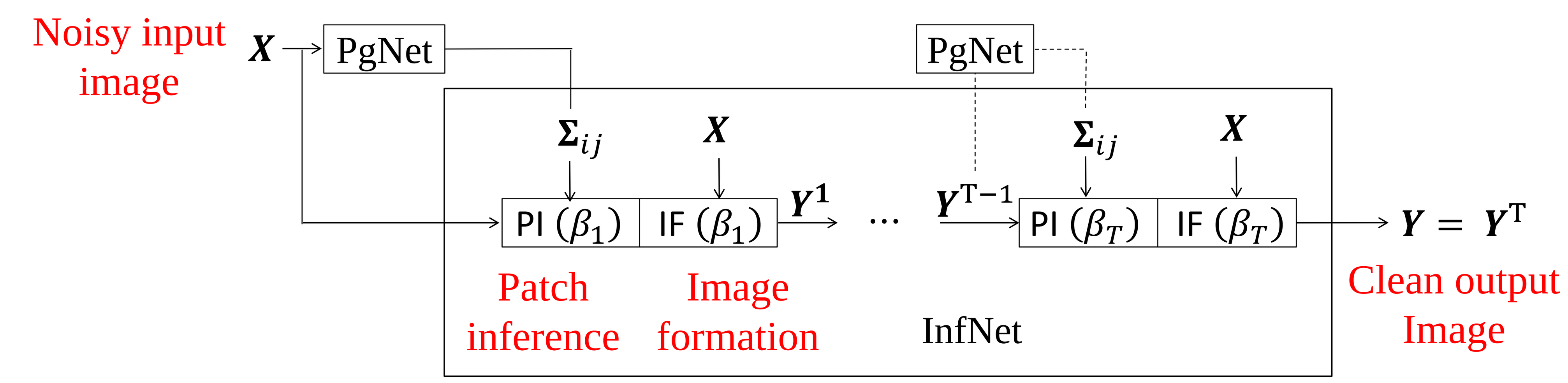
- The proposed deep network consists of two components:

  Parameter generation network (PgNet): Takes the image $\boldsymbol{X}$ as input and generates the pairwise potential function parameters $\boldsymbol{\Sigma}_{ij}(\boldsymbol{x}_{ij})$.



  Quadratic layer: $\boldsymbol{s}_{ij}^k = -\frac{1}{2}\bar{\boldsymbol{x}}_{ij}^T(\boldsymbol{W}_k + \sigma^2\boldsymbol{I})^{-1}\bar{\boldsymbol{x}}_{ij}$

  Inference network (InfNet): Performs Gaussian CRF inference using the parameters $\boldsymbol{\Sigma}_{ij}$.



Experimental Results

- 400 training images (200 from BSD300 and 200 from PASCAL VOC 2012)
- 8×8 image patches, 6 inference steps, 200 Ψ_k matrices, L-BFGS for training
- Trained two networks, one for low noise levels ($\sigma \leq 25$) and one for high noise levels ($\sigma \geq 25$)
- Used $\sigma = [8, 13, 18, 25]$ to train low noise network and $\sigma = [30, 35, 40, 50]$ to train high noise network

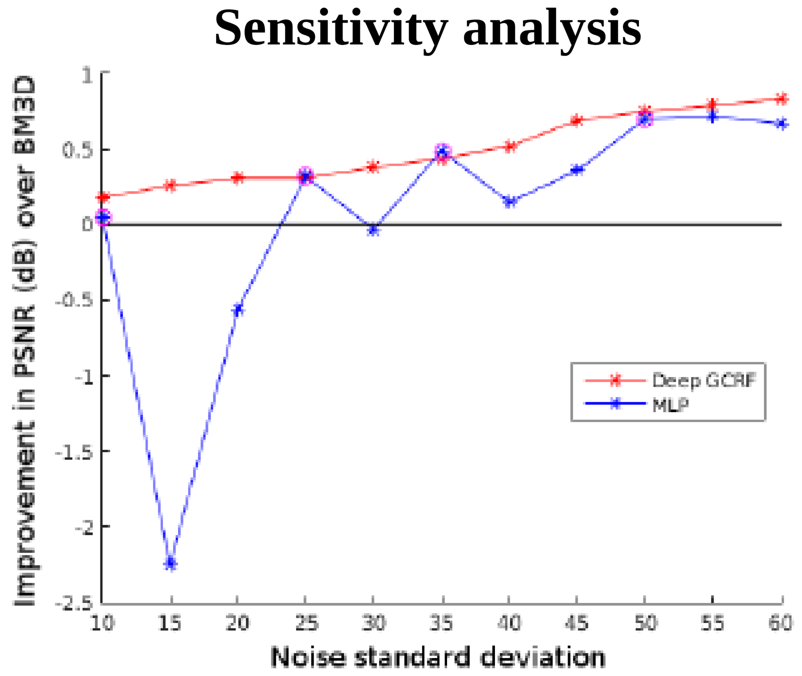
Average PSNR values for 68 images from BSD300 dataset under the unquantized setting

Test σ	ARF	LLSC	EPLL	opt-MRF	ClusteringSR	NCSR	BM3D	MLP	WNNM	CSF	RTF_5	TRD	DGCRF_8
15	30.70	31.27	31.19	31.18	31.08	31.19	31.08	-	31.37	31.24	-	**31.43**	**31.43**
25	28.20	28.70	28.68	28.66	28.59	28.61	28.56	28.85	28.83	28.72	28.75	**28.95**	**28.89**

Average PSNR values for 68 images from BSD300 dataset under the quantized setting

Test σ	LLSC	EPLL	opt-MRF	ClusteringSR	NCSR	BM3D	NL-Bayes	MLP	WNNM	CSF	RTF_5	DGCRF_8
15	31.09	31.11	31.06	30.93	31.13	31.03	31.06	-	31.20	-	-	**31.36**
25	28.24	28.46	28.40	28.26	28.41	28.38	28.43	**28.77**	28.48	28.53	**28.74**	28.73

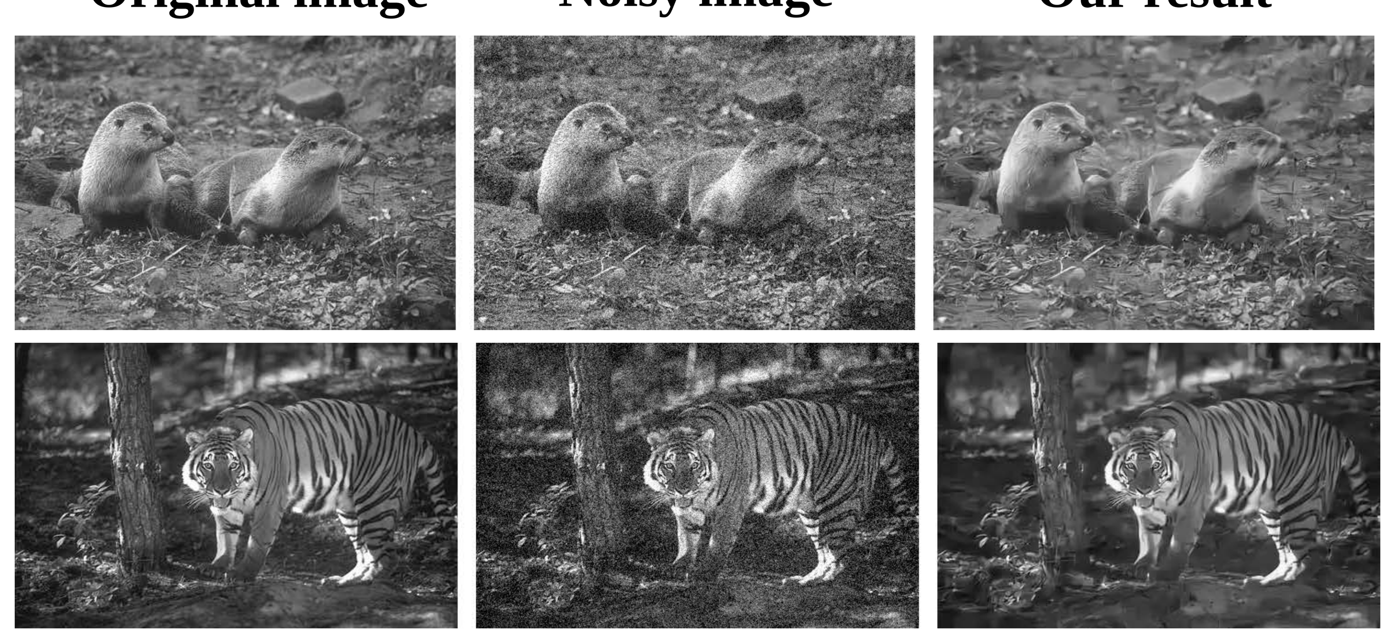
Sensitivity analysis



Comparison of MLP [2] and the proposed approach using a test set of 300 quantized images (100 from BSD300 and 200 from PASCAL VOC 2012) . The noise levels for which MLP was trained are shown using a circular marker.

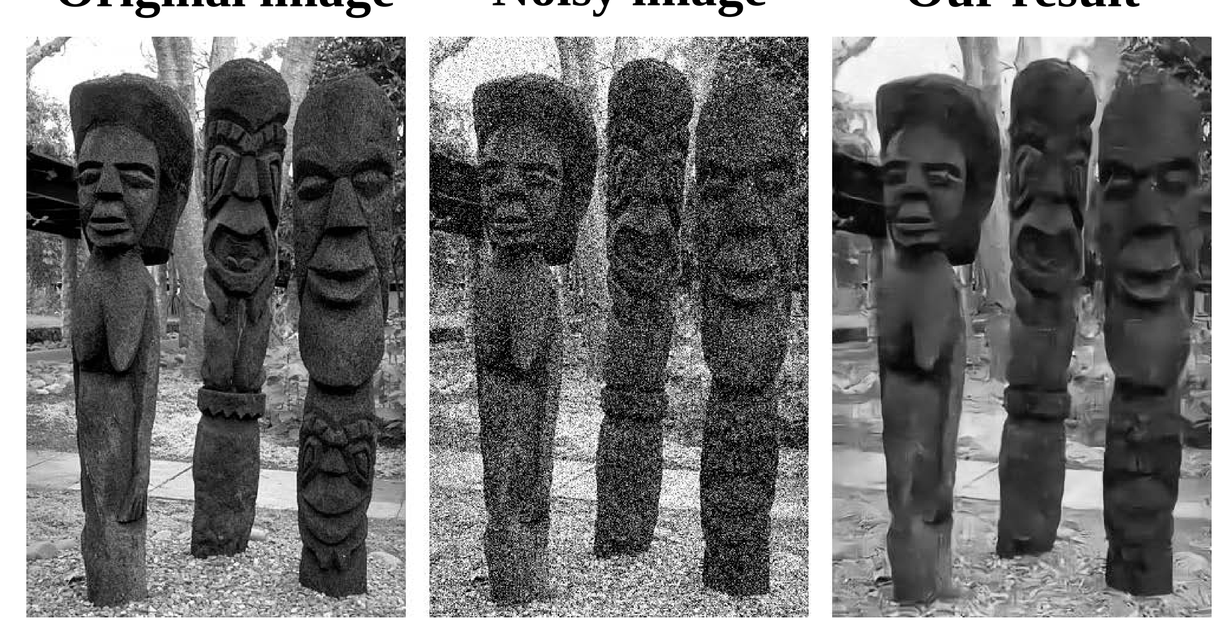
Noise standard deviation $\sigma = 25$

Original image **Noisy image** **Our result**



Noise standard deviation $\sigma = 50$

Original image **Noisy image** **Our result**



References

1. D. Zoran and Y. Weiss. From Learning Models of Natural Image Patches to Whole Image Restoration. In ICCV, 2011.
2. H. C. Burger, C. J. Schuler, and S. Harmeling. Image Denoising: Can Plain Neural Networks Compete with BM3D? In CVPR, 2012.
3. Y. Chen, W. Yu, and T. Pock. On Learning Optimized Reaction Diffusion Processes for Effective Image Restoration. In CVPR, 2015