Modelling and Learning Approaches to Image Denoising

PhD defense talk

Harold Christopher Burger

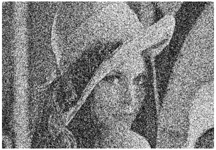
March 18 2013

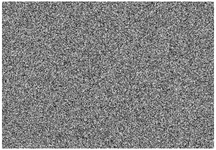
MAX-PLANCK-GESELLSCHAFT

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

Table of Contents

Introduction
What is image denoising?





$$y = x + n$$

y	=	x	+	n
observation		true image		noise

Goal:
Given a noisy observed image, find the noise-free true image.

Image denoising is a decomposition problem.

Introduction
Why is image denoising important?

Image denoising is of growing importance, because of:

1. A flood of data

   Every day approximately 3×10^8 images are uploaded to Facebook alone. This number is growing.

2. The omnipresence of noise

   Images are invariably corrupted by noise. Some sources of noise:

   - read-out noise (Gaussian)
   - dark-current noise (Gaussian)
   - photon shot noise (Poisson)
   - ... many more.

3. Fixed acquisition processes

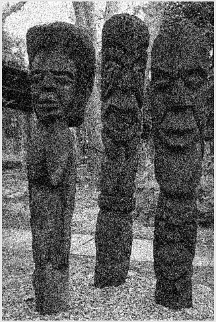
   Modifying the image acquisition process so as to reduce noise is often not possible.

Introduction
An Example

unknown true image

x

noisy observation



y

$\sigma = 50$

PSNR $=$ 15.09dB

$\rightarrow$

denoising result

$\hat{x}$

PSNR $=$ 22.64dB

An increase in PSNR indicates better results.

Introduction
Two denoising paradigms

We divide denoising approaches into two paradigms:

1. Make sophisticated assumptions about **image statistics**

   Assume “AWG” noise: Additive, white Gaussian noise, with uniform variance. **A lot of research** uses this paradigm.

2. Make sophisticated assumptions about **noise statistics**

   Make few assumptions about image statistics.
   Less common.

Introduction
Two denoising paradigms

First paradigm

Approaches placing emphasis on understanding images can be further divided into:

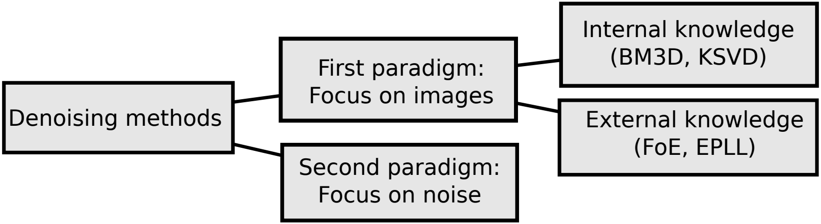
1. Approaches using “internal” image priors: The model adapts to the noisy image at hand (K-SVD, BM3D). **Until recently [1], the best denoising methods were part of this category**.
2. Approaches using “external” generic image priors (FoE, EPLL)

Second paradigm

Understanding the properties of camera sensor noise by studying dark-frames.

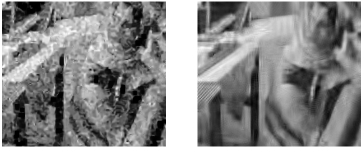
[1] **Image denoising: Can plain neural networks compete with BM3D?** **H.C. Burger**, C.J. Schuler, and S. Harmeling. CVPR 2012.

Introduction
Method taxonomy

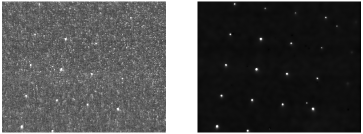


Introduction: Talk Overview

1. Improving existing approaches using a meta-procedure



2. Denoising astronomical images using noise statistics (focus on noise)



3. State-of-the-art image denoising with machine learning (focus on images, with "external" prior)

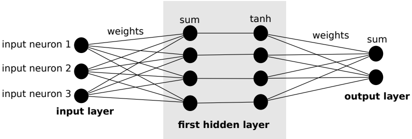


Table of Contents

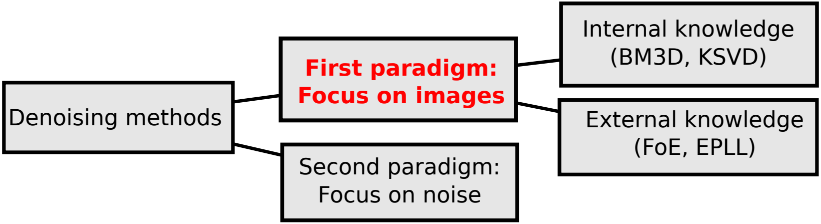
Part 1: Multi-scale denoising

Focus on images

This section summarizes the following publication:

Title:	**Improving denoising algorithms via a multi-scale meta-procedure**
Authors:	**H.C. Burger**, and S. Harmeling
Venue:	Proceedings of the 33rd international conference on Pattern recognition (DAGM). 2011.
Prize:	**This paper was awarded with the DAGM 2011 Prize.**

Part 1: Multi-scale denoising



Part 1: Multi-scale denoising: Motivation

noisy
PSNR: 14.77dB

denoised with KSVD
PSNR: 25.35dB

Low frequency artifacts

Part 1: Multi-scale denoising: Introduction

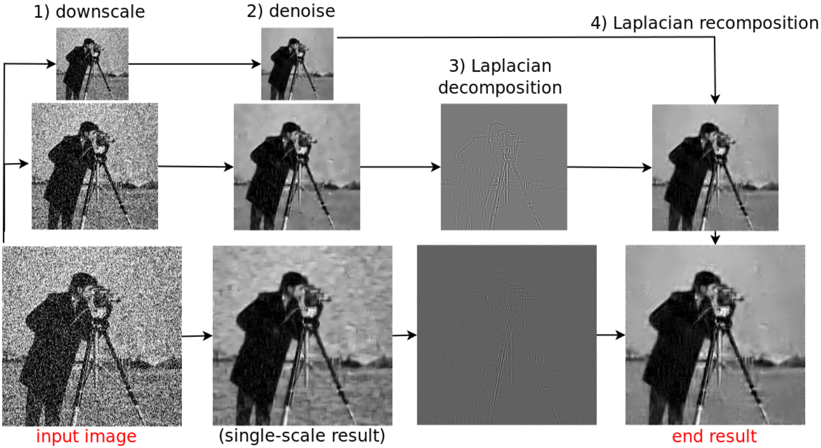
- Hypothesis:

  Most denoising algorithms are best suited for recovering fine-scale information.
- Assumption:

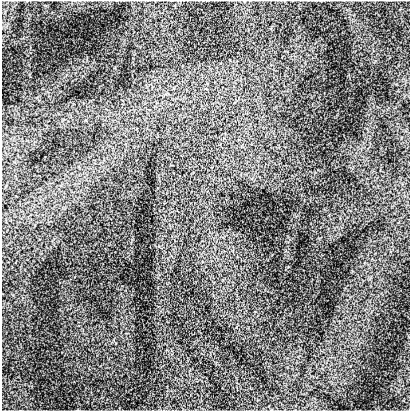
  Statistics of natural images are invariant to changes in spatial scale.
- Contribution:

  A meta-procedure that can be used in combination with existing denoising methods, yet often improves the results. The improvements are largest at high noise levels.

Part 1: Multi-scale denoising
Method: Laplacian pyramids



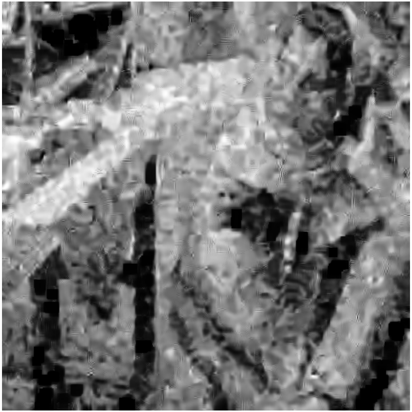
Part 1: Multi-scale denoising: Visual evaluation (1)



noisy $\sigma = 200$
PSNR: 7.59dB

ground truth

Part 1: Multi-scale denoising: Visual evaluation (2)



denoised with BM3D
PSNR: 18.88dB

denoised with MS-BM3D
PSNR: 20.96dB
our result

Part 1: Multi-scale denoising: Conclusions

Conclusions and contributions

- When the noise is high, low frequencies are corrupted, but most methods are bad at recovering them.
- Our method addresses this problem and improves the results in many cases.
- Limitation: Cannot improve algorithms that are already designed to be multi-scale.

Table of Contents

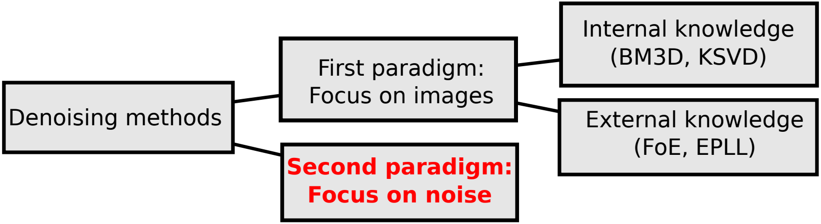
Part 2: Astronomical image denoising

Focus on noise

This section summarizes the following publication:

Title:	**Removing noise from astronomical images using a pixel-specific noise model**
Authors:	**H.C. Burger**, B. Schölkopf, and S. Harmeling
Venue:	IEEE International Conference on Computational Photography (ICCP). 2011.

Part 2: Astronomical image denoising



Part 2: Astronomical image denoising
Astronomical image examples (1)

source: `http://www.pa.uky.edu/~jnorce/ast192.html`

Part 2: Astronomical image denoising
Astronomical image examples (2)

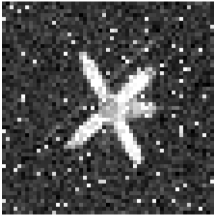
source:
http://www.astronomy-pictures.net/star_clusters.html

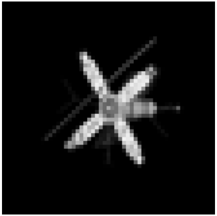
Part 2: Astronomical image denoising
Introduction

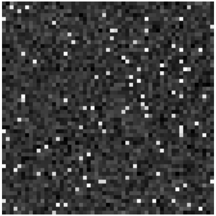
- Assumption:
  Sensor noise is **not** AWGN: **Dark-current** noise due to long exposure times.
- Hypothesis:
  Exploiting statistics of each individual pixel of a sensor leads to better denoising results
- Contribution:
  A denoising method combining a pixel-specific noise model and an image prior adapted to astronomical images.

We consider **dark-current noise**, an important noise component in long-exposure photographs.

Part 2: Astronomical image denoising. Dark-frames







$y = x + n$

observation | true image | dark-current

Focus on images:

- x has interesting structure
- n is AWGN

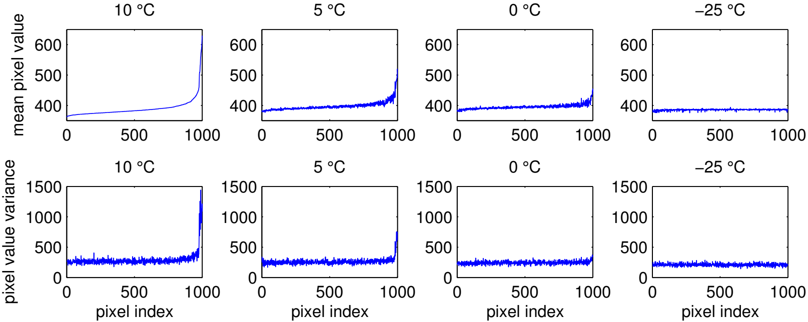
Focus on noise:

- we assume little about x
- n has interesting structure

Dark-current can be recorded with a closed shutter: a "**dark-frame**".

Part 2: Astronomical image denoising
Dark-frame properties

Our assumption was that the noise is not AWG.



The pixels are not equally noisy. Our assumption is justified.

Part 2: Astronomical image denoising
Method DF-MAP$_p$

Method principle:

- Each pixel is modeled with a Gaussian distribution.
- Neighboring pixels should be similar.

We write down the log-posterior for x:

$$-\log p(x|y) = -\log p(y|x) - \log p(x) + c,$$

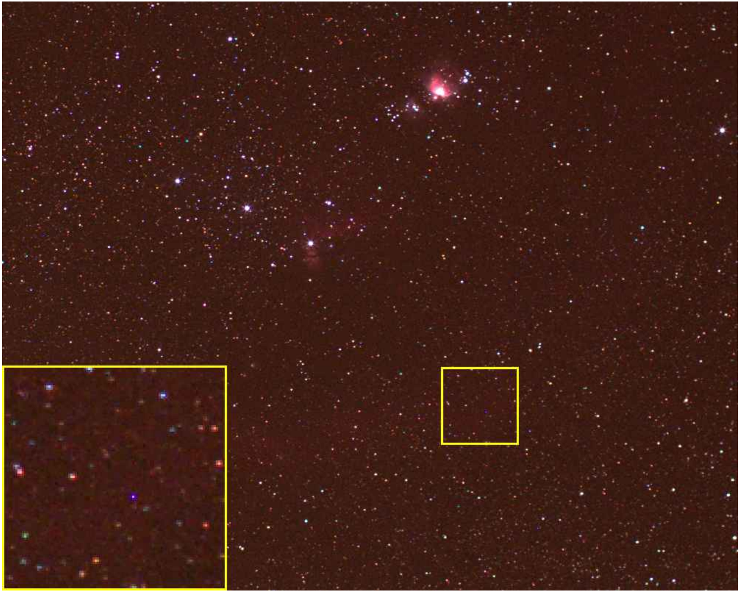
Log-likelihood of y: $\quad -\log p(y|x) = \sum_i \frac{(y_i - x_i - \mu_i)^2}{2\sigma_i^2} + c$.

Prior over x: $\quad -\log p(x) = \lambda \frac{1}{|N_i|} \sum_{j \in N_i} |x_i - x_j|^p + c$.

We minimize the log posterior $-\log p(x|y)$ with gradient descent steps.

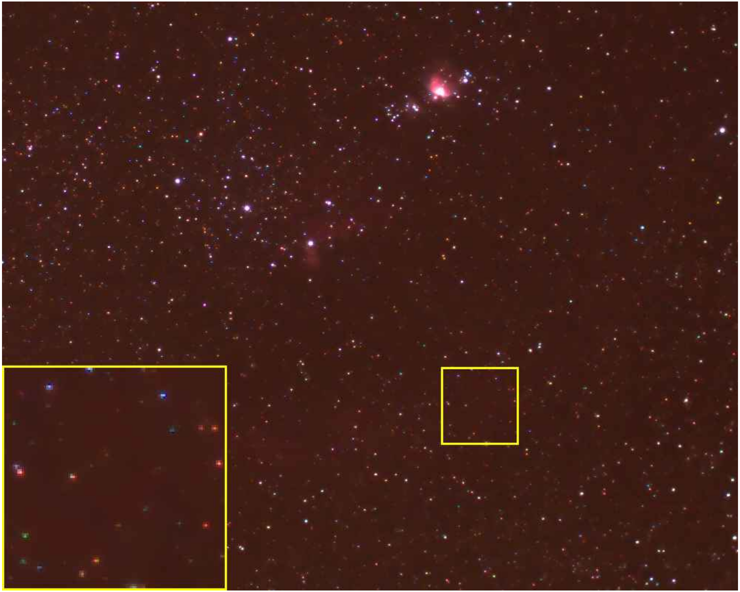
DF-MAP$_p$ method

Part 2: Astronomical image denoising. Results, Orion (1)



Noisy

Part 2: Astronomical image denoising. Results, Orion (2)



DF-MAP$_{1.4}$

Part 2: Astronomical image denoising
Conclusion

Conclusions and Contributions

- Pixel-specific statistical description of the noise
- An image prior adapted to astronomical images
- A simple optimization procedure

Table of Contents

Part 3: Image denoising with neural networks

Focus on images, exploiting "external" prior knowledge

This section summarizes the following publications:

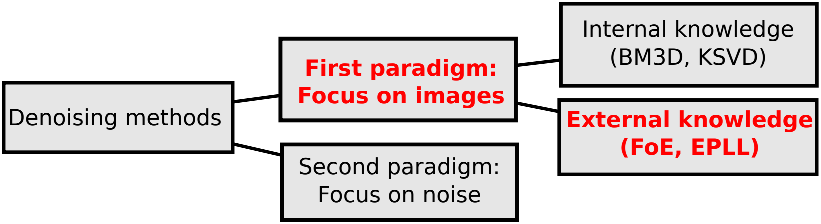
Title:	**Image denoising: Can plain neural networks compete with BM3D?**
Authors:	**H.C. Burger**, C.J. Schuler, and S. Harmeling
Venue:	IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). 2012.

Part 3: Image denoising with neural networks

Title:	**Image denoising with multi-layer perceptrons, part1: Comparison with existing algorithms and with bounds**
Authors:	**H.C. Burger**, C.J. Schuler, and S. Harmeling
Submitted to:	The Journal of Machine Learning Research (JMLR). 2012.
Pre-print:	arXiv:1211.1544

Title:	**Image denoising with multi-layer perceptrons, part2: Training trade-offs and analysis of hidden activation patterns**
Authors:	**H.C. Burger**, C.J. Schuler, and S. Harmeling
Submitted to:	The Journal of Machine Learning Research (JMLR). 2012.
Pre-print:	arXiv:1211.1552

Part 3: Image denoising with neural networks



Part 3: Denoising with Neural Networks: Motivation

- Engineering vs. learning:
  BM3D and other state-of-the-art denoising methods are heavily engineered.
  Q: Is it possible to achieve good results with a learning-based method?

- Generic vs. internal image priors:
  Generic image priors should theoretically be able to yield good results.
  Q: Can we find a practical procedure?

YES!
Our learning-based approach outperforms all competing methods.

Part 3: Denoising with Neural Networks: What are Neural Networks?

Multi-layer perceptrons:

A multi-layer perceptron (MLP) is a nonlinear function that maps vector-valued input via several hidden layers to vector-valued output.

Example:

$$f(x) = b_3 + W_3 \tanh(b_2 + W_2 \tanh(b_1 + W_1 x)).$$

Given labeled data, one can learn the set of parameters $\theta = \{W_1, W_2, W_3, b_1, b_2, b_3\}$

Part 3: Denoising with Neural Networks.
Applying a learned MLP

How to denoise with a learned MLP:

- The MLP is applied patch-wise ("sliding-window" manner).
- Patches are treated independently.
- We average in areas where patches overlap.

Part 3: Denoising with Neural Networks
Learning to denoise

Learning: We train MLPs to learn the mapping from noisy patches to clean patches: $\hat{x} = f(y) = f(x + n)$, using stochastic gradient descent.

What is new about this?

- We use a **large training set** (ImageNet, $\approx 6 \times 10^6$ images)
- We choose MLPs with **large capacity** (up to four hidden layers, 2047 hidden units per layer).
- We use **large patch sizes** (39×39 and 17×17)

This was made possible through the use of GPUs.

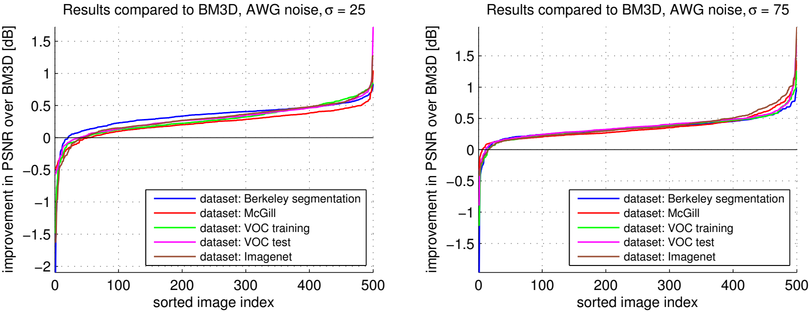
Part 3: Denoising with Neural Networks
Progress during training

Part 3: Denoising with Neural Networks
Training insights

Insights regarding the training procedure

- **No overfitting** due to abundance of training data.
- More (varied) training data always helps.
- There is a trade-off between **capacity** and **training time**. Regarding **capacity**:
  - More hidden units always help.
  - There is an ideal number of hidden layers. Too many hidden layers cause difficult optimization.
- Larger input patches help.
- There is an ideal size for the output patch.
- Fine-tuning (reducing the learning rate at the end) helps.

Part 3: Denoising with Neural Networks
Results: Performance profiles



For σ=25 and σ=75, the MLPs outperform BM3D (which is considered to be the best or one of the best methods) on 92.1% and 97.6% of the images, respectively.

Part 3: Denoising with Neural Networks
Results on other noise types: Stripes


14.68 dB


BM3D: 24.38 dB


MLP: **30.11** dB

Part 3: Denoising with Neural Networks
Results on other noise types: Salt-and-pepper


12.41 dB


median filter: 30.33 dB


MLP: **35.08** dB

Part 3: Denoising with Neural Networks
Results on other noise types: JPEG artifacts

27.33 dB

SA-DCT[2]: 28.96 dB

MLP: **29.42** dB

[2] **Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images.** A. Foi, V. Katkovnik, and K. Egiazarian. IEEE Transactions on Image Processing (TIP). 2007.

Part 3: Denoising with Neural Networks
Results on other noise types: Poisson noise

2.87 dB

GAT+BM3D[3]: 22.90 dB

MLP: **24.26** dB

[3] **Optimal inversion of the generalized Anscombe transformation for Poisson-Gaussian noise.** M. Mäkitalo and A. Foi. IEEE Transactions on Image Processing (TIP). 2012.

Part 3: Denoising with Neural Networks
Comparison to bounds

Recent work estimates ultimate bounds in denoising quality:

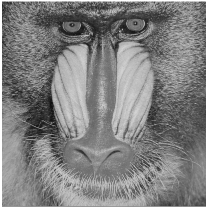
1. "Clustering-based" bounds:
   - **Is denoising dead?** P. Chatterjee and P. Milanfar, IEEE Transactions on Image Processing (TIP), 2010
2. "Bayesian patch-based" bounds:
   - **Natural Image Denoising: Optimality and Inherent Bounds**, A. Levin, and B. Nadler, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011
   - **Patch complexity, finite pixel correlations and optimal denoising**, Levin, A. and Nadler, B. and Durand, F. and Freeman, W.T., European Conference on Computer Vision (ECCV), 2012

Theoretical bounds are often compared to BM3D because of its excellent performance.

Part 3: Denoising with Neural Networks
Comparison to bounds (1)

1. "Clustering-based" bounds[4]:

(images taken from the paper[4])



bounds[4], $\sigma = 25$:	25.61dB	28.94dB
results with MLP, $\sigma = 25$:	**26.01**dB	**29.25**dB

We can outperform these bounds.

[4]**Is denoising dead?** P. Chatterjee and P. Milanfar, IEEE Transactions on Image Processing (TIP), 2010

Part 3: Denoising with Neural Networks
Comparison to bounds (2)

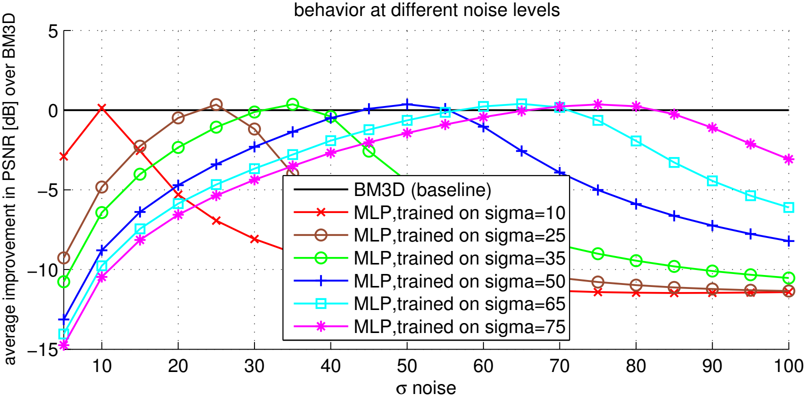
2. "Bayesian patch-based" bounds[5]

	bounds[5] vs. BM3D	MLPs vs. BM3D
$\sigma = 10$	-	0.07dB
$\sigma = 25$	-	0.3dB
$\sigma = 35$	0.6dB	0.33dB
$\sigma = 50$	0.7dB	0.34dB
$\sigma = 65$	-	0.40dB
$\sigma = 75$	1.0dB	0.38dB
$\sigma = 170$	-	2.19dB

- Our method outperforms BM3D on all noise levels.
- We make important progress toward reaching the bounds.

[5] **Patch complexity, finite pixel correlations and optimal denoising**, Levin, A. and Nadler, B. and Durand, F. and Freeman, W.T., European Conference on Computer Vision (ECCV), 2012

Part 3: Denoising with Neural Networks
Limitations



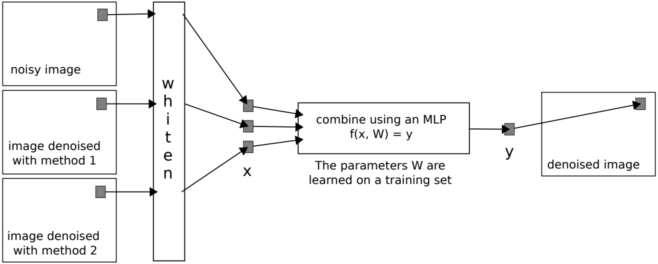
The MLPs have to be trained on each noise level individually.

Part 3: Denoising with Neural Networks
Results: "Easy" and "hard" images

The hard images have repeating structure.

Part 3: Denoising with Neural Networks
Combining Neural Networks with BM3D

Q: Can we get the strenghts of BM3D and of neural networks by combining their results?



We combine the results of BM3D and of an MLP using a second MLP ("E-MLP").

Part 3: Denoising with Neural Networks: E-MLPs

A: YES!
The results are usually better than the best of the inputs.

easy

BM3D:	29.10dB
MLP:	*29.98*dB
E-MLP:	**30.25**dB

hard

BM3D:	*26.02*dB
MLP:	25.57dB
E-MLP:	**26.18**dB

YES! Better results overall:

	bounds[6] vs. BM3D	MLPs vs. BM3D	E-MLPs vs. BM3D
$\sigma = 10$	-	0.07dB	**0.15**dB
$\sigma = 25$	-	0.3dB	**0.38**dB
$\sigma = 35$	0.6dB	0.33dB	**0.45**dB
$\sigma = 50$	0.7dB	0.34dB	**0.52**dB
$\sigma = 75$	1.0dB	0.38dB	**0.53**dB
$\sigma = 170$	-	2.19dB	**2.32**dB

[6] **Patch complexity, finite pixel correlations and optimal denoising**, Levin, A. and Nadler, B. and Durand, F. and Freeman, W.T., ECCV, 2012

Part 3: Denoising with Neural Networks
Understanding

We achieved outstanding image denoising performance with MLPs.

But:
How do the MLPs work?

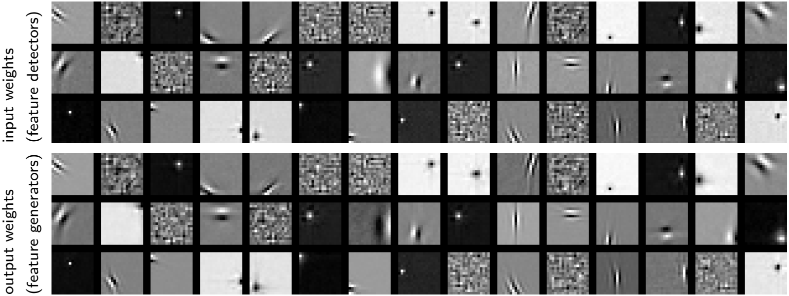
Understanding the functioning principle of our MLPs seems impossible at first.

Two tools will help us:

- Analyzing input and output weights.
- Finding the input pattern maximizing the activation of a given hidden unit.

Part 3: Denoising with Neural Networks
Understanding: Feature detection/generation

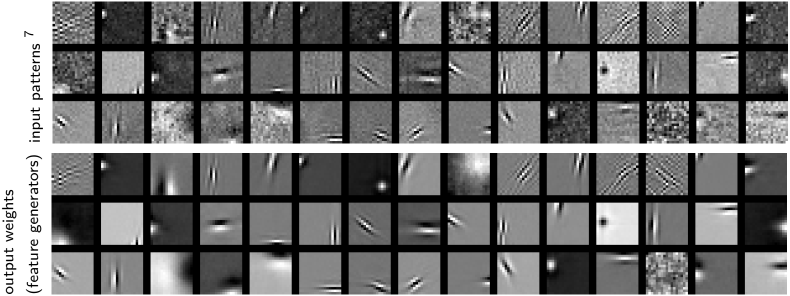
MLP with a single hidden layer $(17, 2047, 17)$:



If a feature is detected, the same feature is copied to the output.

Part 3: Denoising with Neural Networks
Understanding: Feature detection/generation

MLP with four hidden layers $(17, 4 \times 2047, 17)$:

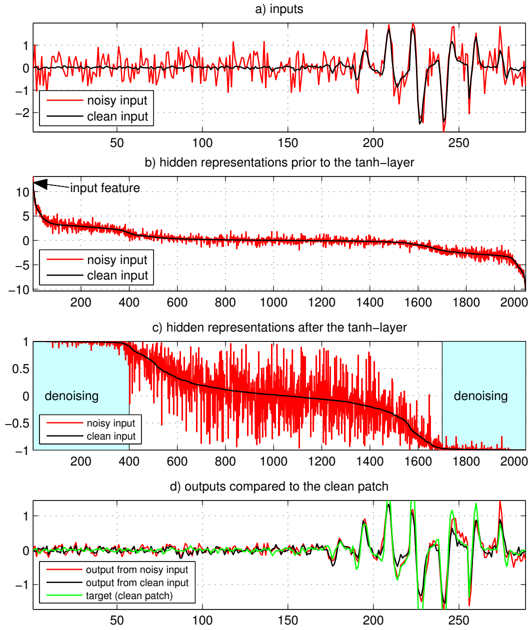


Input patterns are found via activation maximization[7]

If a feature is detected, the same feature is copied to the output.

[7] **Understanding Representations Learned in Deep Architectures.** Erhan, D. and Courville, A. and Bengio, Y.,Technical Report 1355, Université de Montréal/DIRO. 2010

Part 3: Understanding. Noise removal via saturation



Part 3: Denoising with Neural Networks
Understanding

How do the MLPs work?

Key insights:

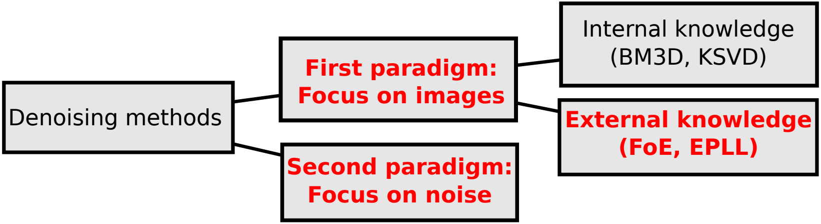
1. Noise is attenuated through saturation.
2. Image information is preserved due to the high activation values of the corresponding feature detectors/generators.

Part 3: Denoising with Neural Networks
Conclusion

- We were able to achieve state-of-the-art image denoising performance using MLPs:
  - Best performance of all denoising algorithms.
  - Can beat clustering-based bounds.
  - Getting close to Bayesian patch-based bounds.
- We achieve good results on other types of noise.
- Understanding denoising MLPs: MLPs detect features and generate the same features. Noise is removed via saturation.
- Limitations:
  1. MLPs have to be trained on each noise level individually.
  2. MLPs do not reach state-of-the-art performance on images with repeating structure.

Table of Contents

Conclusion: Thesis Contributions



Conclusion: Thesis Contributions

Image denoising is a long-standing problem.

Three contributions were presented:

- Part 1 How to improve existing methods at high noise levels.
- Part 2 How to denoise in the setting where the noise has structure.
- Part 3 How to achieve state-of-the-art denoising results with a learning-based approach.

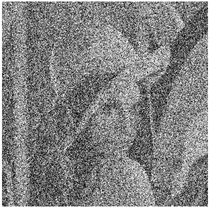
End of talk.

Appendices

Appendix overview

- Multi-scale denoising
- Astronomical image denoising
- MLPs:
  - MLPs
  - BM-MLPs
  - E-MLPs
  - Other restoration tasks with MLPs?
    - Deconvolution with MLPs
    - Others
  - Other architectures?
- Miscellaneous
  - Potential future work
  - Deep learning
  - Repeated application

Multi-scale denoising: Denoising by down-scaling



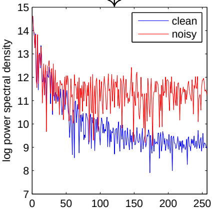
→

→

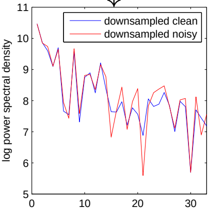
7.61 dB

21.04 dB

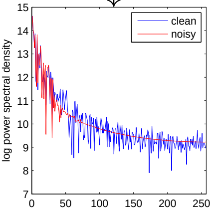
20.17 dB



noisy

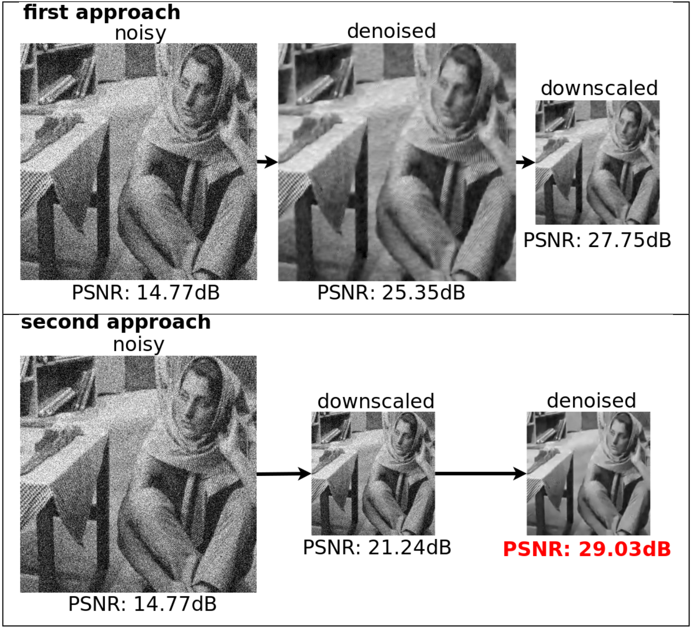


down-sampled

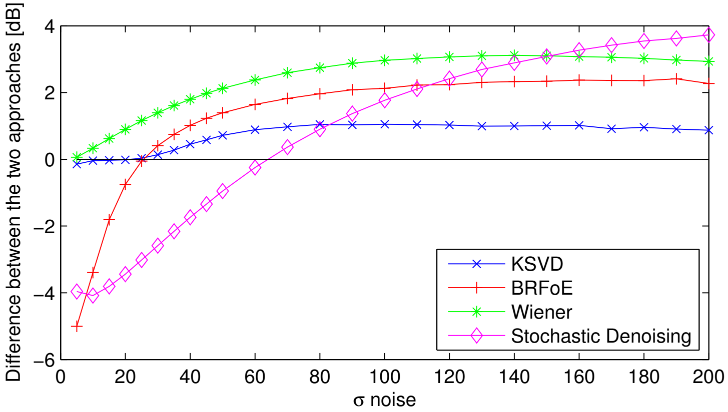


up-sampled

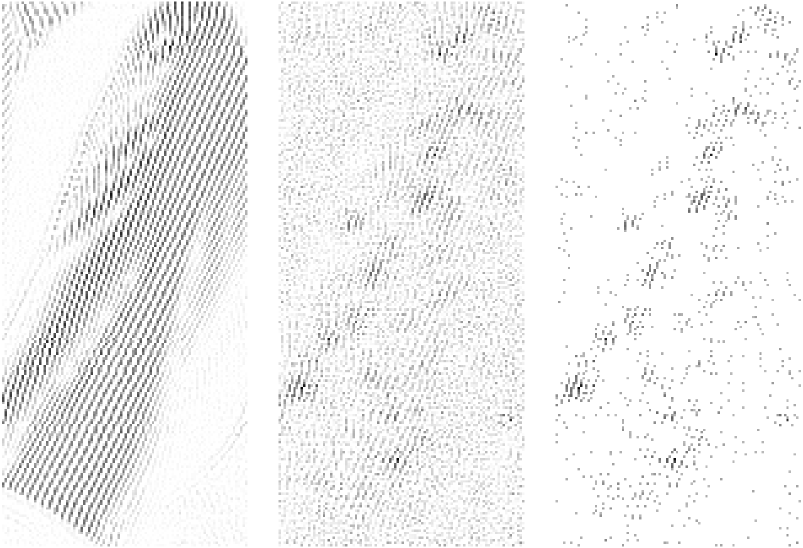
Multi-scale denoising: Denoising lower frequencies (1)



Multi-scale denoising: Denoising lower frequencies (2)

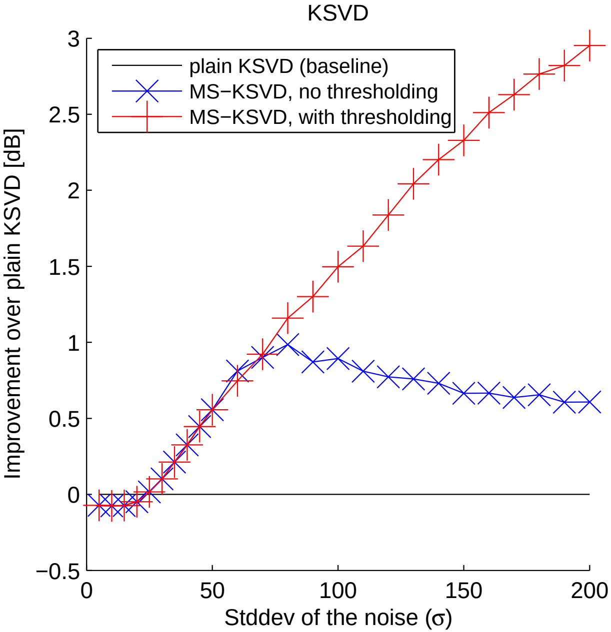


Part 1: Multi-scale denoising: Thresholding



true denoised thresholded

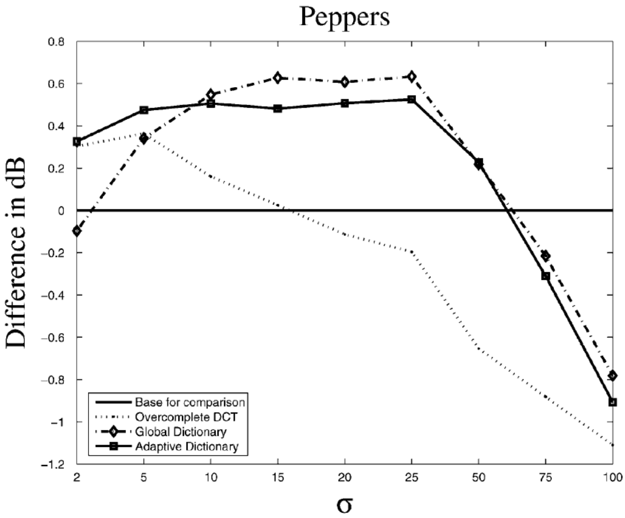
Multi-scale denoising: Thresholding (2)



Multi-scale denoising: Training images

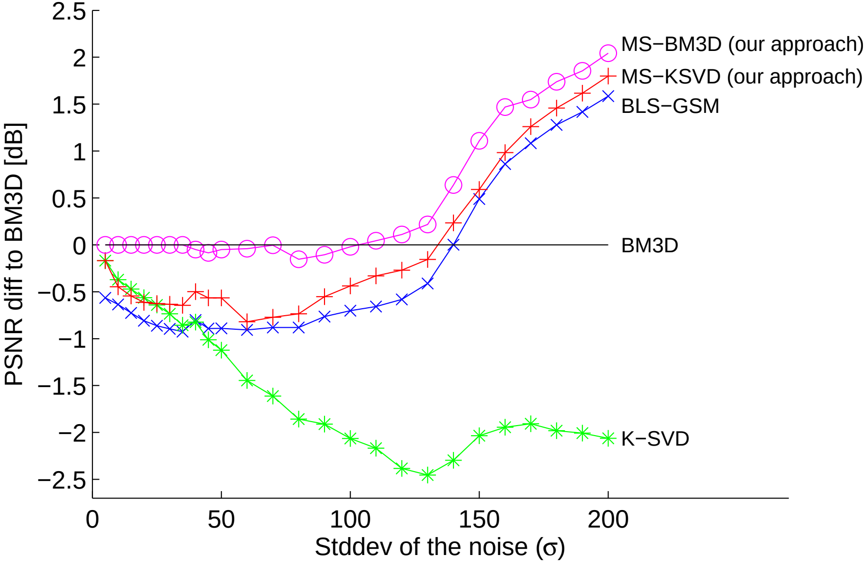
Multi-scale denoising: Test images

Part 1: Multi-scale denoising: KSVD vs. BLSGSM [8]



[8] **Image denoising via sparse and redundant representations over learned dictionaries**. Elad, M. and Aharon, M. IEEE Transactions on Image Processing (TIP), 2006

Part 1: Multi-scale denoising: All vs. BM3D



Multi-scale denoising: Related work

Estrada's method:
"Multi-pass" denoising [1] to handle "large scale" noise

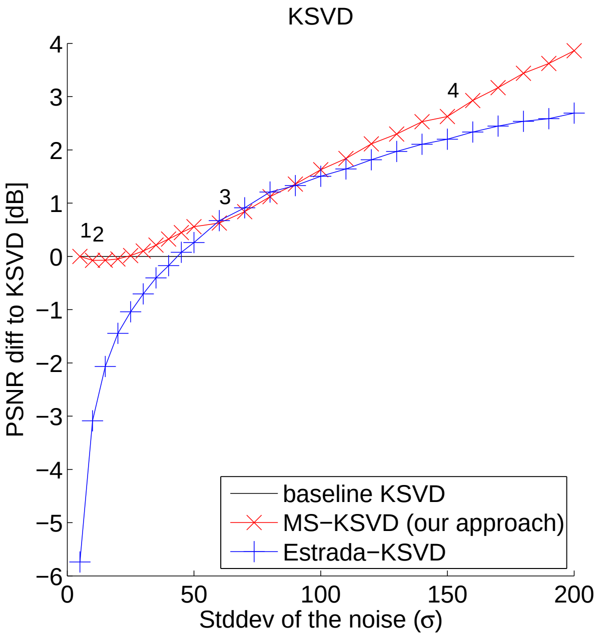
$$I_{final}(x,y) = \alpha(x,y) I^*(x,y) + (1 - \alpha(x,y)) I^*_{hu}(x,y), \qquad (1)$$

where

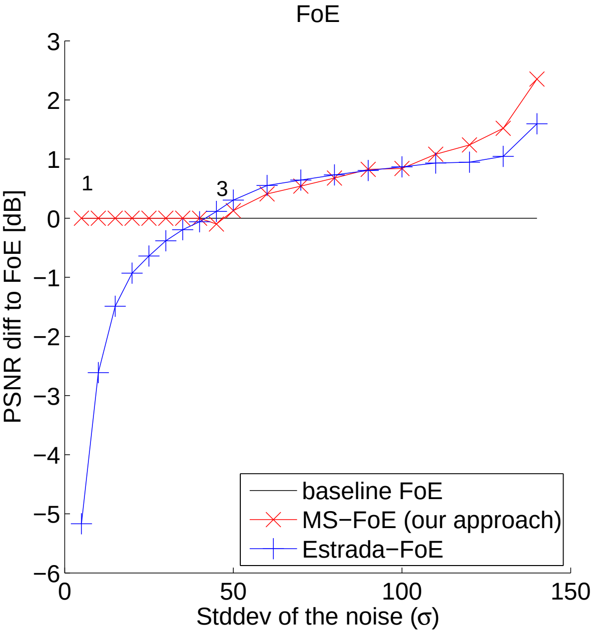
$$\alpha(x,y) = |\nabla I^*|. \qquad (2)$$

Intuition:
Preserve sharp detail where the high resolution image has edge structure. For more uniform regions, prefer a denoised estimate computed at a coarser scale.

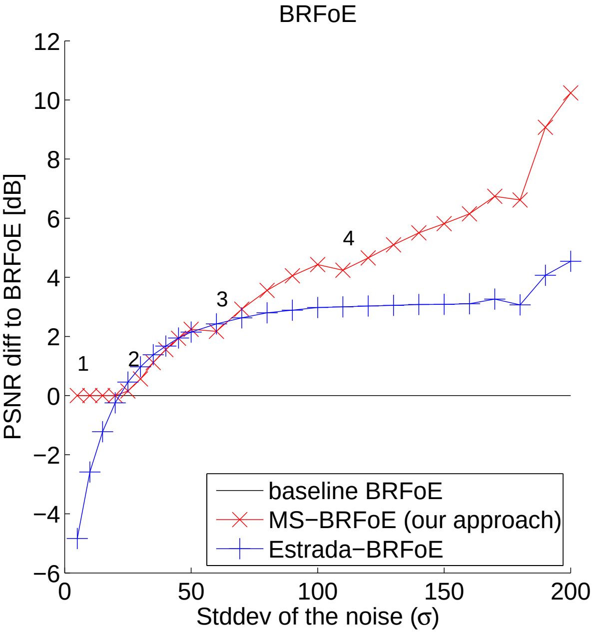
Multi-scale denoising: Results (1)



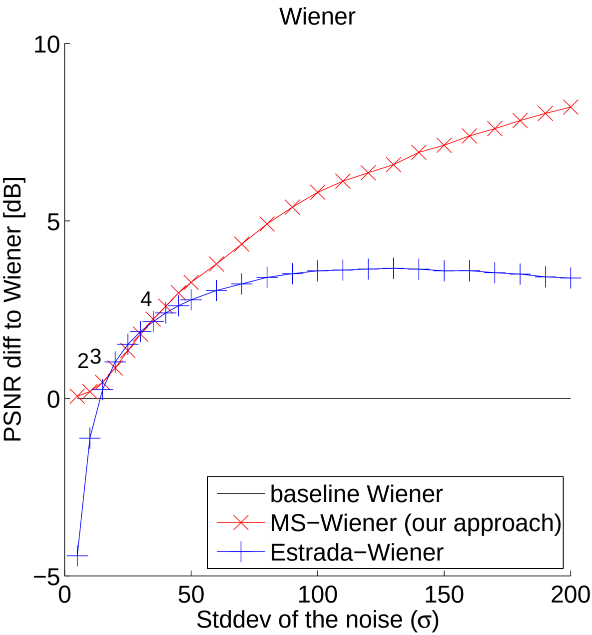
Multi-scale denoising: Results (2)



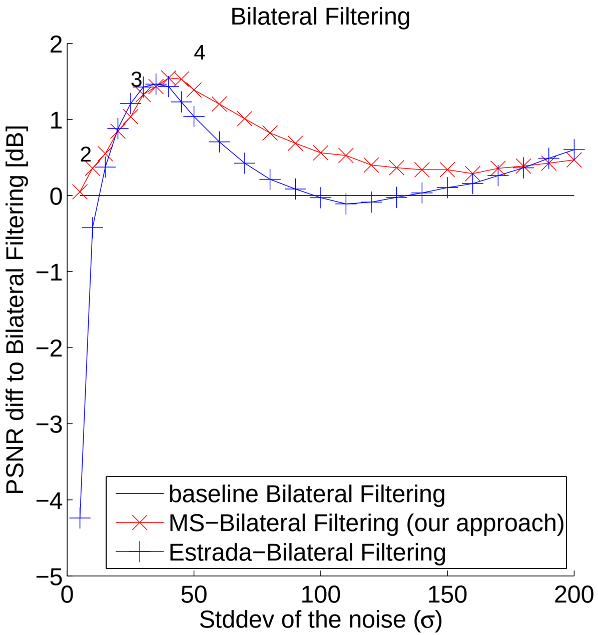
Multi-scale denoising: Results (3)



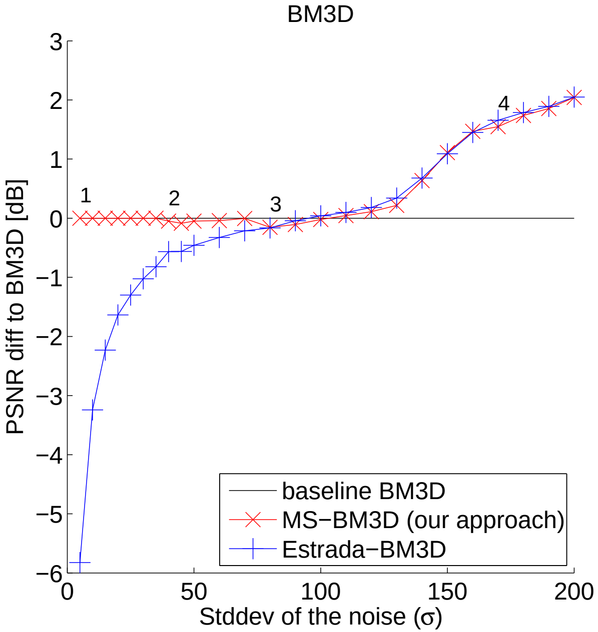
Multi-scale denoising: Results (4)



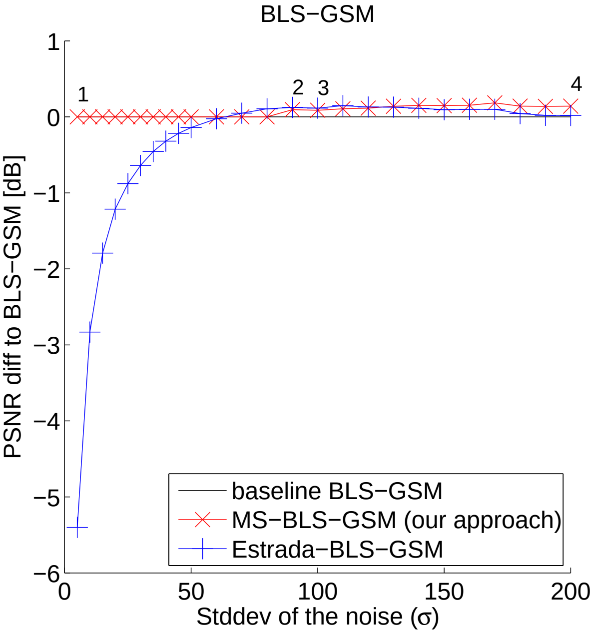
Multi-scale denoising: Results (5)



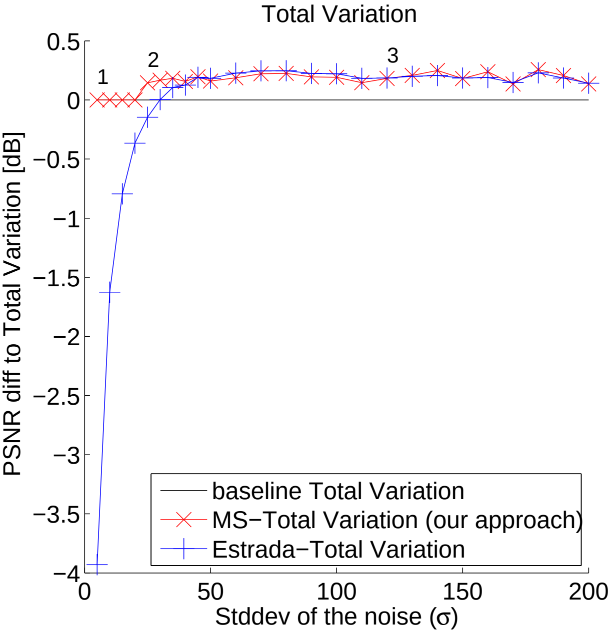
Multi-scale denoising: Results (6)



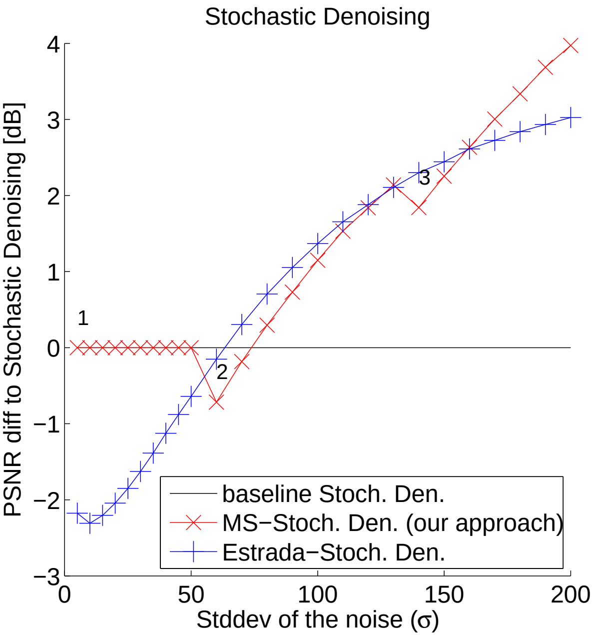
Multi-scale denoising: Results (7)



Multi-scale denoising: Results (8)



Multi-scale denoising: Results (9)



Jump to Appendix

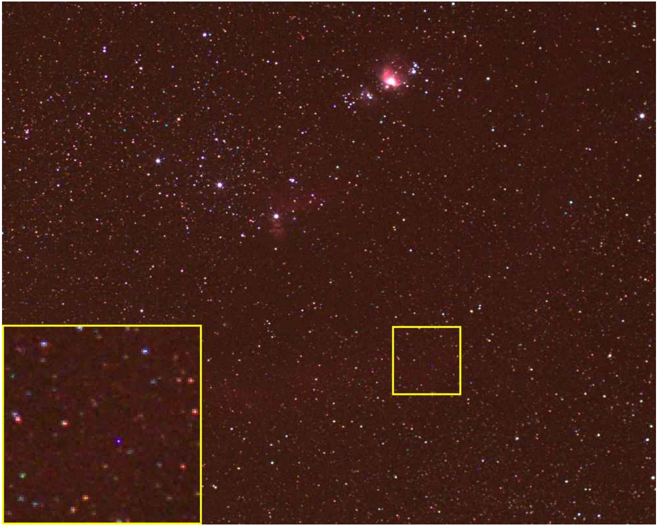
Dark-frame denoising
Related work

An approach by Manuel Gomez-Rodriguez et al. [2]:

- Assumes a library of dark-frames is given
- Attempts to minimize the discrete gradient of the image at some pixels
- Creates an artificial dark-frame that is a convex combination of a subset of dark-frames in the library.

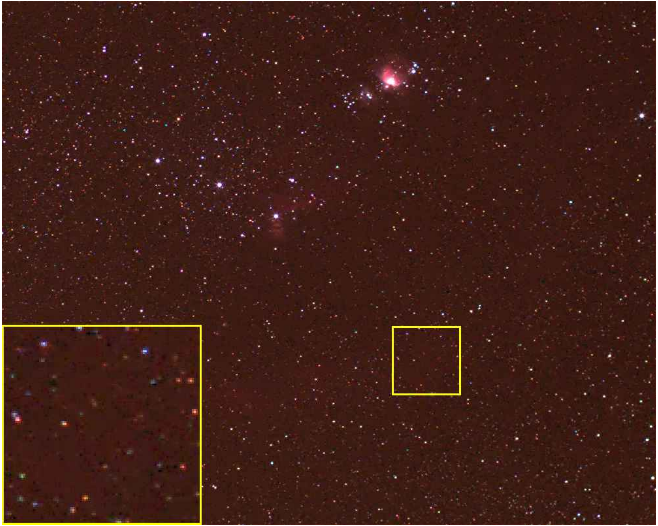
Solving this problem involves a **QP**.

Dark-frame denoising
Results, Orion (1)



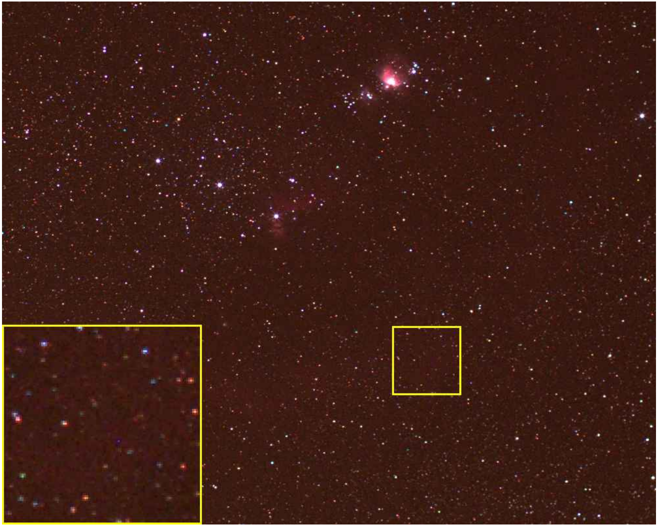
Noisy

Dark-frame denoising
Results, Orion (2)



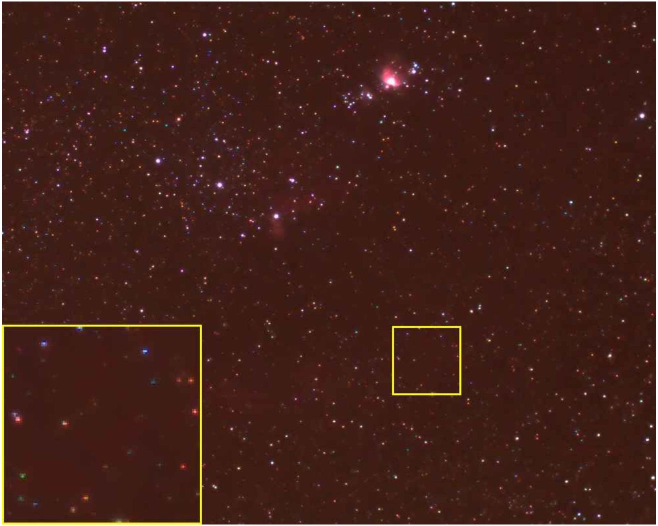
BLS-GSM

Dark-frame denoising
Results, Orion (3)



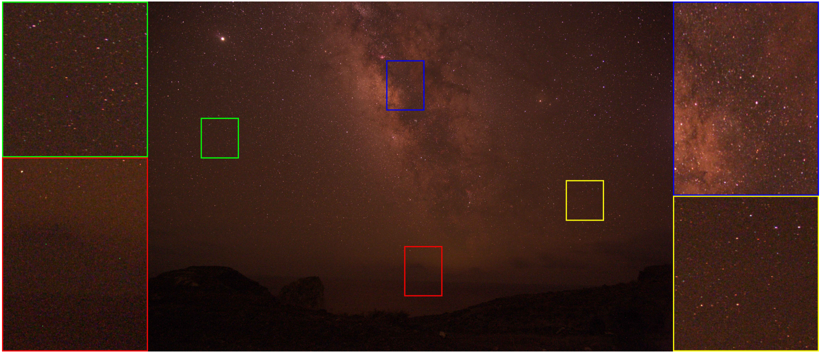
QP

Dark-frame denoising
Results, Orion (4)



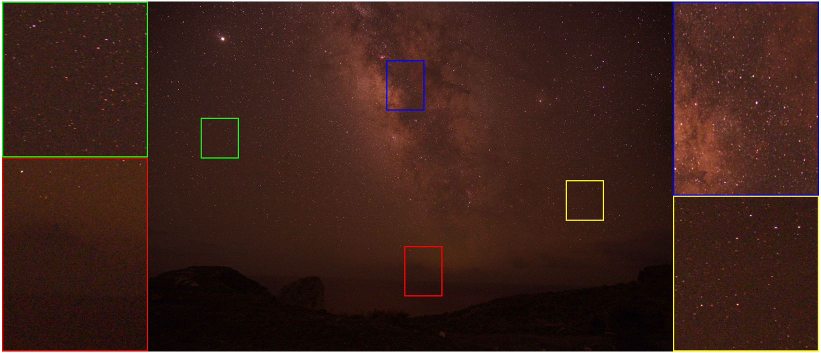
DF-$MAP_{1.4}$

Dark-frame denoising
Results, Milky Way (1)



Noisy

Dark-frame denoising
Results, Milky Way (2)



QP

Dark-frame denoising
Results, Milky Way (3)

DF-MAP$_{1.4}$

Jump to Appendix

Denoising with Neural Networks: More "easy" images


MLP vs. BM3D: +0.89dB


MLP vs. BM3D: +0.86dB


MLP vs. BM3D: +0.86dB


MLP vs. BM3D: +0.84dB


MLP vs. BM3D: +0.82dB


MLP vs. BM3D: +0.82dB

Images where the MLP outperforms BM3D, for $\sigma = 25$.

Denoising with Neural Networks: More “hard” images (1)

MLP vs. BM3D: -2.09dB

MLP vs. BM3D: -1.03dB

MLP vs. BM3D: -0.75dB

Images where BM3D outperforms the MLP, for $\sigma = 25$.

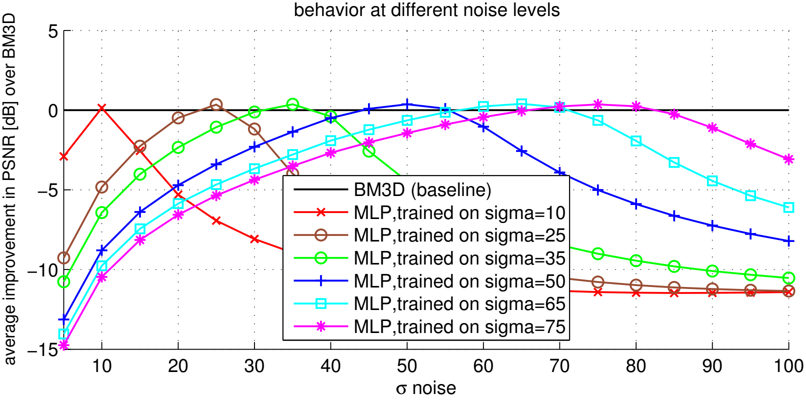
Denoising with Neural Networks: More “hard” images (2)

MLP vs. BM3D: -1.09dB

MLP vs. BM3D: -0.66dB

MLP vs. BM3D: -0.54dB

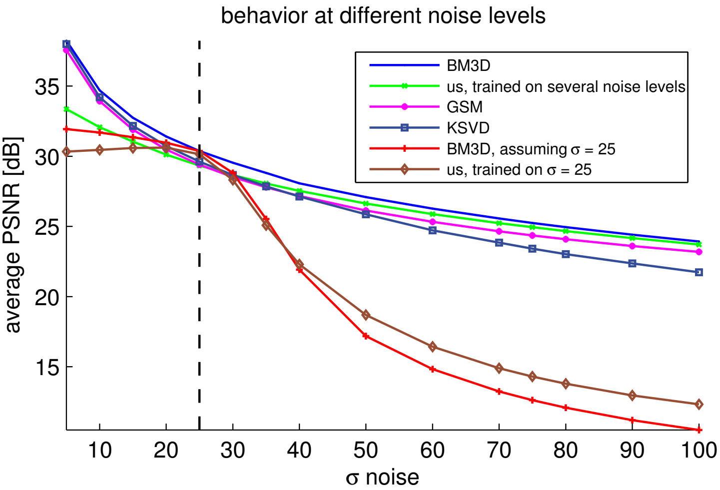
Denoising with Neural Networks: Limitations



The MLPs have to be trained on each noise level individually.

Denoising with Neural Networks: Limitations: Possible solution

We tried to train an MLP on several noise levels.



Denoising with Neural Networks: Other limitations

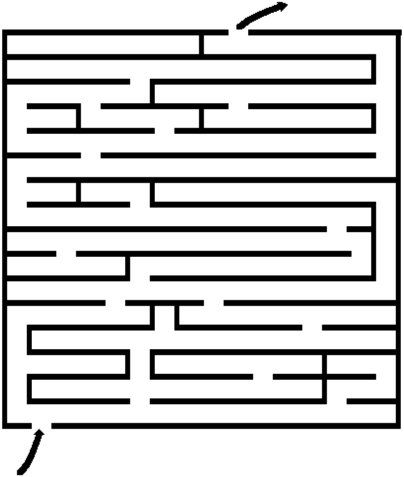
Are there other limitations?

- Handling large-scale noise (e.g. wide bands)
- Not clear if possible to handle other quality measures (e.g. SSIM), because we currently assume patches to be independent. (We could optimize the patch-wise SSIM).

Jump to Appendix

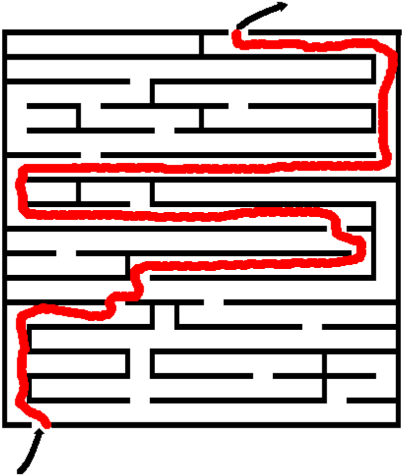
BM-MLPs: Intuition (1)

Can neural networks learn arbitrary mappings?



BM-MLPs: Intuition (2)

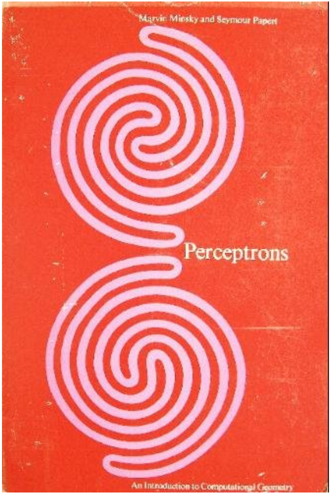
Can neural networks learn arbitrary mappings?



Probably not.

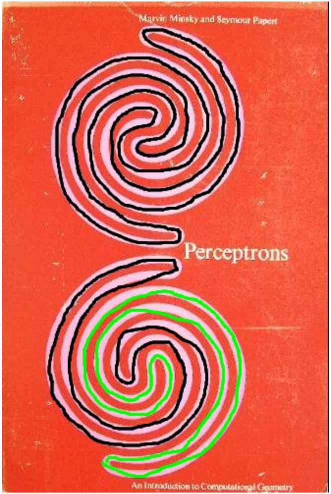
BM-MLPs: Intuition (3)

Can neural networks learn arbitrary mappings?



BM-MLPs: Intuition (4)

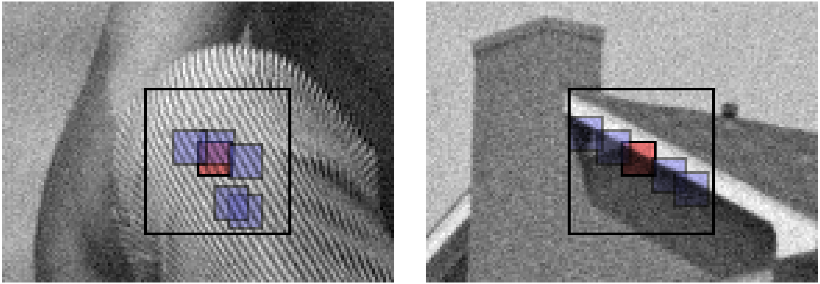
Can neural networks learn arbitrary mappings?



Probably not.

BM-MLPs: Intuition (5)

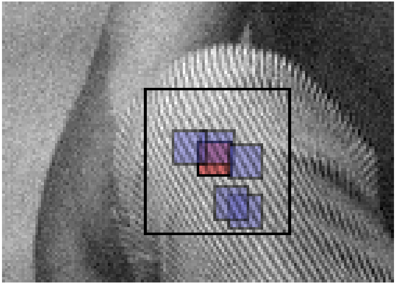
BM3D is effective on repetitive images due to block matching:

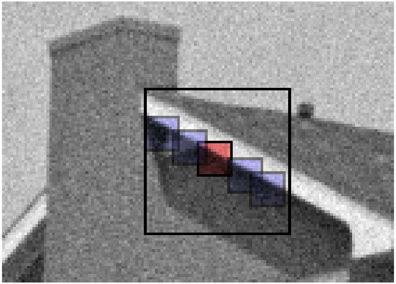


Block-matching might be difficult to learn with a feed-forward architecture.

Part 3: Denoising with Neural Networks
Why are some images hard to denoise?

BM3D is effective on repetitive images due to block matching:





Q: Can we achieve better results by combining block matching with neural networks?

Part 3: Denoising with Neural Networks
Neural networks combined with block matching: Results

A: Combining block matching with neural networks does not help much.

image	BM3D	NLSC	MLP	BM-MLP
Barbara	**30.67**dB	*30.50*dB	29.52dB	29.75dB
Boat	29.86dB	29.86dB	**29.95**dB	*29.92*dB
C.man	29.40dB	29.46dB	*29.60*dB	**29.67**dB
Couple	29.68dB	29.63dB	**29.75**dB	*29.73*dB
F.print	**27.72**dB	27.63dB	*27.67*dB	27.63dB
Hill	29.81dB	29.80dB	*29.84*dB	**29.87**dB
House	*32.92*dB	**33.08**dB	32.52dB	32.75dB
Lena	32.04dB	31.87dB	**32.28**dB	*32.17*dB
Man	29.58dB	29.62dB	*29.85*dB	**29.86**dB
Montage	**32.24**dB	*32.15*dB	31.97dB	32.11dB
Peppers	30.18dB	*30.27*dB	30.27dB	**30.53**dB

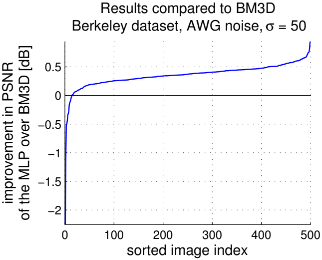
Block-matching MLP compared to plain MLPs and other algorithms for $\sigma = 25$.

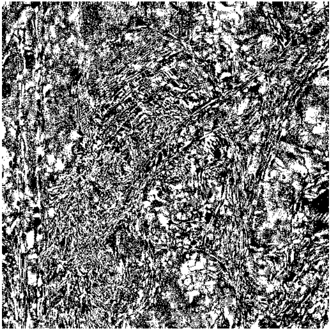
BM-MLPs: Where do BM-MLPs help?

The MLP with block-matching outperforms the plain MLP on this image. Regions where the block-matching MLP is better are highlighted.

Jump to Appendix

E-MLPs: Justification





No method is always the best. **Left:** On average, MLPs outperform BM3D on the Berkeley segmentation dataset. However, on some images, the MLP is much worse than BM3D. No method is the best on all images. **Right:** Pixels in image "Lena" where BM3D is worse than an MLP are white, pixels where BM3D is better are black. No method is the best on all parts of the image.

Part 3: Denoising with Neural Networks
Combining Neural Networks with BM3D

image	BM3D	NLSC	MLP	E-MLP: MLP and BM3D
Barbara	**27.21**dB	*27.13*dB	25.37dB	26.95dB
Boat	26.72dB	26.73dB	*27.02*dB	**27.11**dB
C.man	26.11dB	26.36dB	*26.42*dB	**26.75**dB
Couple	26.43dB	26.33dB	*26.71*dB	**26.78**dB
F.print	*24.53*dB	24.25dB	24.23dB	**24.57**dB
Hill	27.14dB	27.05dB	*27.32*dB	**27.40**dB
House	29.71dB	*29.88*dB	29.52dB	**30.00**dB
Lena	28.99dB	28.88dB	*29.34*dB	**29.46**dB
Man	26.76dB	26.71dB	*27.08*dB	**27.13**dB
Montage	27.69dB	28.02dB	*28.07*dB	**28.34**dB
Peppers	26.69dB	26.73dB	*26.74*dB	**27.18**dB

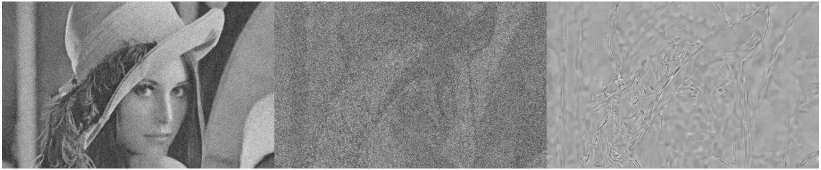
Ensembling BM3D and MLP with an MLP, $\sigma = 50$.

The results are usually better than the best of the two inputs.

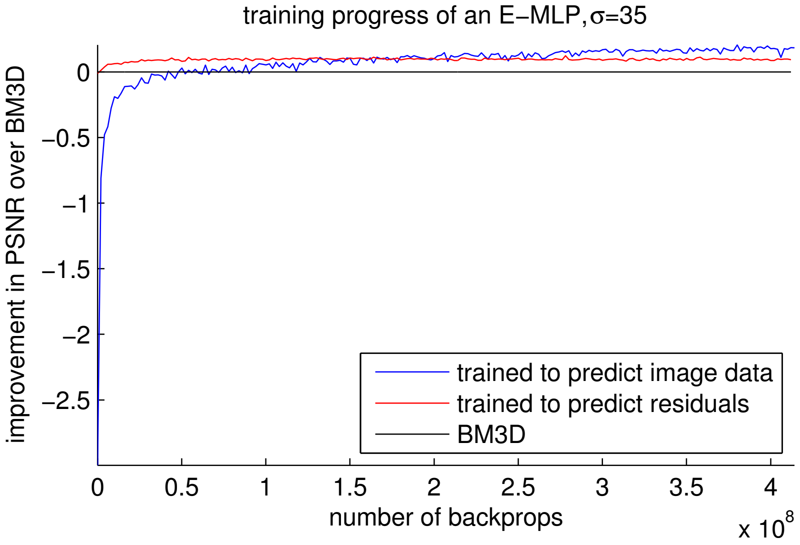
E-MLPs
Whitening

input:

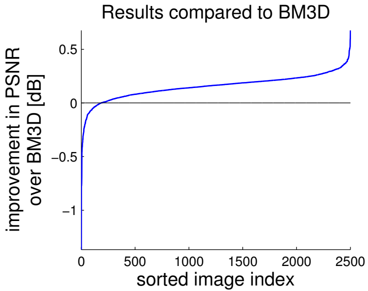
output:

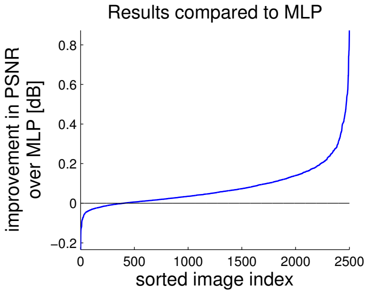


E-MLPs: Train on residuals?

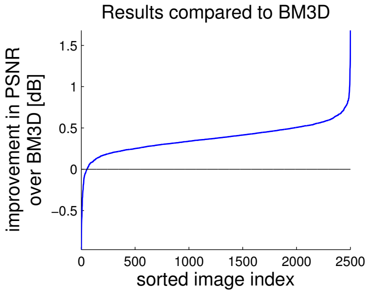


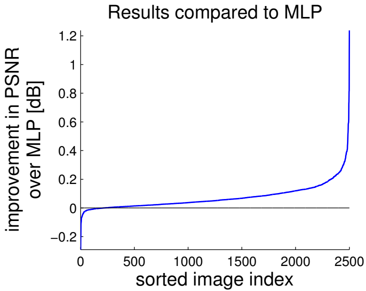
E-MLPs
Results: $\sigma = 10$





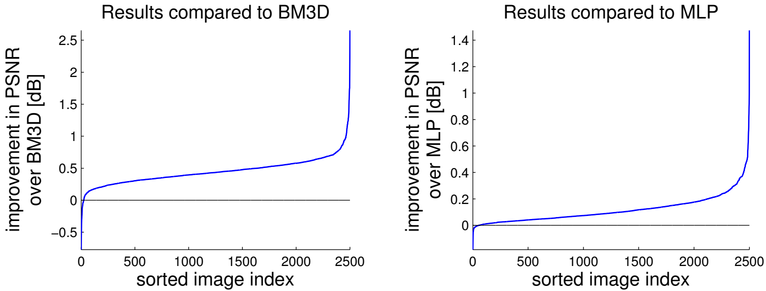
E-MLPs
Results: $\sigma = 25$





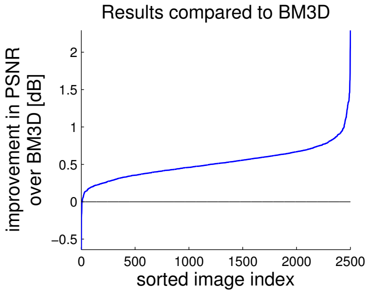
E-MLPs

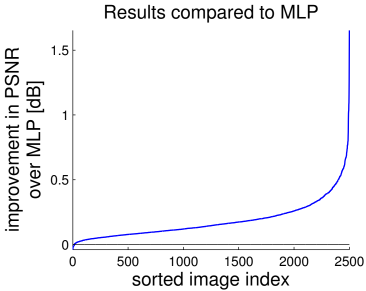
Results: $\sigma = 35$



E-MLPs

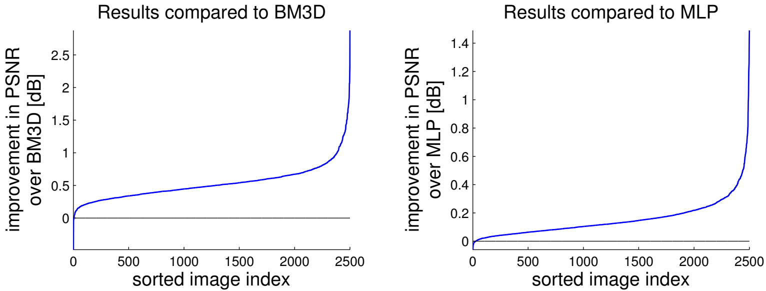
Results: $\sigma = 50$





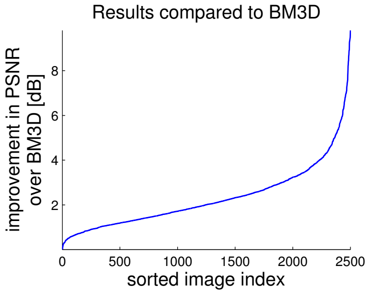
E-MLPs

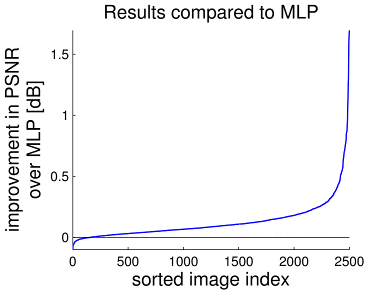
Results: $\sigma = 75$



E-MLPs

Results: $\sigma = 170$



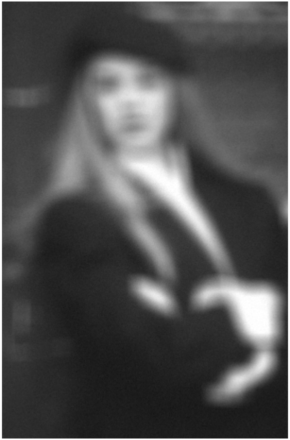


Jump to Appendix

Other restoration tasks with MLPs?

- **Denoising:** Find a clean image, given a noisy one.
  - **“Artifact removal”:** Remove e.g. JPEG-artifacts.
- **Deconvolution:** Find a sharp image, given a blurry one.
- **Super-resolution:** Find a high-resolution image, given a low-resolution image.
- **Inpainting:** Restore missing image content.
- **Demosaicking:** Reverse the effect of the color filter array.

Other restoration tasks with MLPs?
1. Deconvolution



corrupted $\rightarrow$ $\rightarrow$ clean

Other restoration tasks with MLPs?
1. Deconvolution

Deconvolution comes in different flavors:

- **Non-blind, not spatially varying.**
  - In the case "one kernel, one MLP": "Solved" by Schuler et al. [9].
  - Future work: Can one MLP handle multiple (all?) kernels?
- **Non-blind, spatially varying.** New difficulty: Handle the smooth variation of the blur over the image. Use EFF [10]?
- **Blind, not spatially varying.** New difficulty: The MLP has to predict the blur kernel, given the **whole** image.
- **Blind, spatially varying.** Most difficult.

---

[9] **A machine learning approach for image deconvolution**. Schuler, C.J. and **Burger, H.C.** and Schölkopf, B. and Harmeling, S. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2013.

[10] **Efficient filter flow for space-variant multiframe blind deconvolution**, Hirsch, M. and Sra, S. and Schölkopf, B. and Harmeling, S. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

Deconvolution with neural networks

Title:	**A machine learning approach for image deconvolution**
Authors:	C.J. Schuler, **H.C. Burger**, B. Schölkopf and S. Harmeling
Accepted at:	IEEE International Conference on Computer Vision and Pattern Recognition (CVPR). 2013.

Image deconvolution with neural networks: Idea

Problem:

$$y = x * v + n$$

Find x, given y and v.

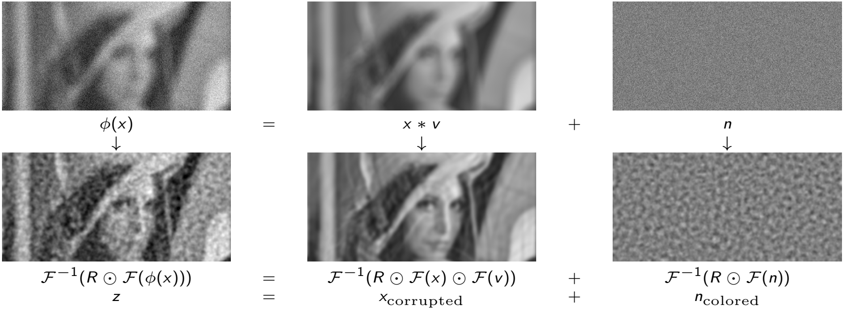


Image deconvolution with neural networks: Results

clean	corrupted	Krishnan et al.	MLP
	20.36 dB	25.81 dB	**27.02** dB

Image deconvolution with neural networks: Results

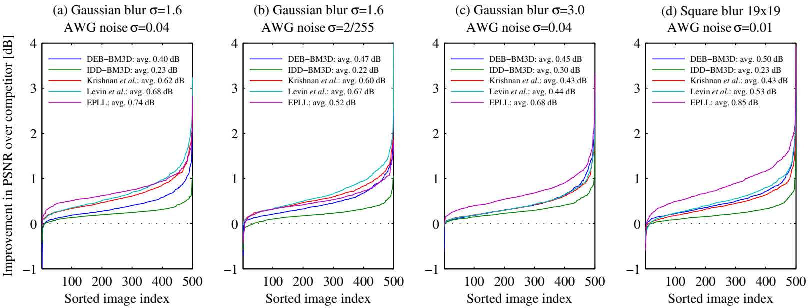


Image deconvolution with neural networks: Results

Defocused Image MLP

Removal of defocus blur in a photograph. The true PSF is approximated with a pillbox.

Jump to Appendix

Other restoration tasks with MLPs?
2. Super-resolution

Super-resolution:

- Naive approach: Small patch comes in, large patch comes out.
- Problem: How to we acquire training data? Specifically, what is the "correct" low-pass filter to create low-resolution images from high-resolution images?
- Potential challenge: High-dimensional outputs are difficult for MLPs.

Other restoration tasks with MLPs?
3. Inpainting

Inpainting comes in different flavors:

- "non-blind": The region to be in-painted is known.
- "blind": The region to be in-painted is not known.

Obtaining training data is probably easy.

Possible challenges:

- Potentially difficult if the region to be inpainted is large. Possible solutions: Multiple in-painting iterations, multi-scale procedure...
- Potentially difficult to identify region to be in-painted (might need a prior over the shape of the region to be inpainted).

Other restoration tasks with MLPs?
4. Demosaicking

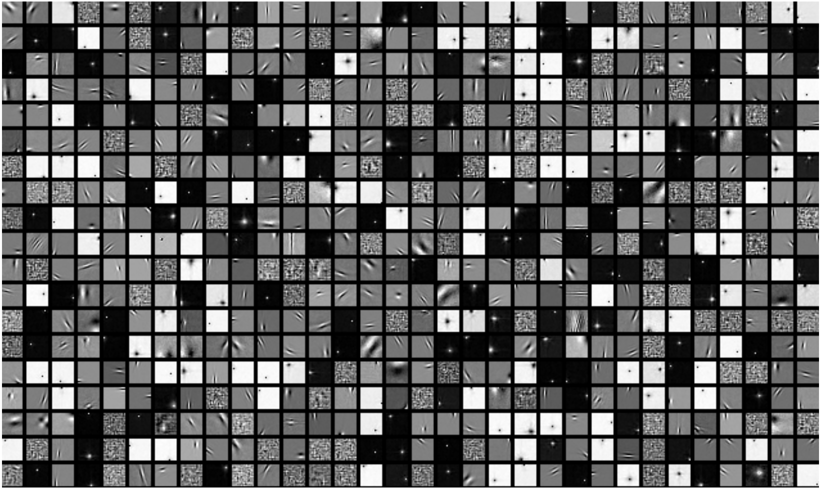
Demosaicking:

- Obtaining training data should be easy.
- Existing algorithms already obtain high PSNR values. Therefore probably difficult to achieve impressive improvements (the high PSNR regime seems difficult for MLPs).

Jump to Appendix

Other architectures? Motivation

Some output weights:



Many features are translated or rotated versions of each other.

Wasteful use of parameters

Other architectures? Potential candidates

Some potential alternatives to plain MLPs:

- Convolutional nets [11]. Already used for denoising [12].
- Tiled convolutional nets [13].
- Sparsity-enforcing machines [14].
- Potentially many others...

---

[11] **Gradient-based learning applied to document recognition.** Yann LeCun et al. 1998.

[12] **Natural image denoising with convolutional networks**, Viren Jain and Sebastian Seung. NIPS 2008

[13] **Tiled convolutional neural networks**, Le, Quoc V et al. NIPS 2010

[14] **A unified energy-based framework for unsupervised learning**, Marc Aurelio Ranzato et al. AISTATS 2007

Other architectures? Trade-offs

CNNs and Tiled CNNs reduce the number of parameters through

1. Local receptive fields, and
2. Parameter sharing.

Pro: Reducing the number of parameters is especially useful when labeled data is scarce.

Pro: CNNs and Tiled CNNs can learn invariances: Some translation invariance for CNNs, some degree of rotation invariance for Tiled CNNs.

Con: Specialized architectures are potentially less powerful than MLPs.

Con: Many choices of architectures exist. Not clear a priori which is best. In that case, it is often best to start with the simplest solution (i.e. MLPs).

Jump to Appendix

Potential future work

- Currently, patches are considered to be independent. This is clearly not ideal. How can we handle patch dependencies? Handling this should improve results (cf. FoE).
- Can we have an MLP that not only denoises well, but also makes a prediction regarding its accuracy? Which parts of the image are denoised well, which are not?
- How can we handle all noise levels with one MLP?
- How can we have a shorter training procedure?
- How can we handle images with repeating structure well?
- Can we optimize other quality measures (e.g. SSIM)?

Jump to Appendix

Deep learning?

What is deep learning?

- Refers to an unsupervised, greedy layer-wise training procedure.
- Usually each layer is trained to reconstruct its input, under some constraints.
- After pre-training, an architecture is fine-tuned on an unrelated supervised task.

Differences and similarities:

- Similarity: Our nets are "deep".
- Similarity: Our nets resemble denoising auto-encoders.
- Difference: One-phase training.
- Difference: Abundance of labeled data.

Deep learning?

Can we benefit from deep learning?

- In [15], RBMs are pre-trained on image data and fine-tuned for image denoising. The results are disappointing compared to our MLPs.
- Our preliminary experiments with stacked denoising auto-encoders are also disappointing.
- Open question: Can we use deep learning to train architectures with more than four hidden layers?

Deep learning is especially useful when labeled data is scarce. We have plenty of labeled data.

[15] **Boltzmann Machines and Denoising Autoencoders for Image Denoising**, Cho, Kyunghyun. arXiv preprint arXiv:1301.3468. 2013.

Jump to Appendix

Repeated application of MLPs

What happens when we apply an MLP on a denoised image?

clean

noisy, PSNR: 20.18dB

first application, PSNR: 32.58dB

second application, PSNR: 30.21dB

Repeated application of MLPs

... And what happens when we continue?

third application, PSNR: 28.74dB

fourth application PSNR: 27.73dB

99th application, PSNR: 11.37dB

100th application, PSNR: 11.35dB

Jump to Appendix

References

F. Estrada, D. Fleet, and A. Jepson.
Stochastic image denoising.
In *Proc. BMVC*, 2009.

M. Gomez-Rodriguez, J. Kober, and B. Schölkopf.
Denoising photographs using dark frames optimized by quadratic programming.
In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–9, 2009.