International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.7, No.3 (2014), pp.171-180
http://dx.doi.org/10.14257/ijsip.2014.7.3.14

Deep Learning for Image Denoising

HuiMing Li

University of Science and Technology Liaoning, Anshan China
Lhm_as@163.com

Abstract

Deep learning is an emerging approach for finding concise, slightly higher level representations of the inputs, and has been successfully applied to many practical learning problems, where the goal is to use large data to help on a given learning task. We present an algorithm for image denoising task defined by this model, and show that by training on large image databases we are able to outperform the current state-of-the-art image denoising methods.

Keywords: *deep learning, image denoising, denoising auto-encoder*

1. Introduction

Image denoising can be described as the problem of mapping from a noisy image to a noisefreeimage. Various methods have been proposed for image denoising. One approach is linear or non-linear filtering methods which are a relatively simple approach based on smoothing, such as Median filtering which replace each pixel with the median of the value of a set of neighboring pixels [1], linear smoothing and wiener filtering. Another one is methods based on wavelet or dictionary decompositions of the image. Wavelet decompositions is to transfer image signals to an alternative domain where they can be more easily separated from the noise, such as BLS-GSM [2]. The dictionary-based method is to denoise by approximating the noisy patch using a sparse linear combination of atoms, including KSVD [3] which is an iterative algorithm that learns a dictionary on the noisy image at hand, NLSC [4] which is one of the best currently available denoising algorithms in terms of quality of the results, but requires long computation times. The last one is methods based on global image statistics or other image properties, such as self-similarities. Typical schemes include EPLL [5] and BM3D [6] which are often considered the state-of-the-art in image denoising.

While these models have been successfully in practice, they share a shallow linear structure. Recent research suggests, however, that non-linear, deep models can achieve superior performance in various real world problems. A few of deep models have also been applied to image denoising [7-11].

Deep learning is an emerging approach within the machine learning research community [12]. Deep learning algorithms have been proposed in recent years to move machine learning systems towards the discovery of multiple levels of representation. Learning algorithms for deep architectures are centered on the learning of useful representations of data, which are better suited to the task at hand, and are organized in a hierarchy with multiple levels. There are several motivations for deep architectures: Brain inspiration (several areas of the brain are organized as a deep architecture); Cognitive arguments and engineering arguments (humans often organize ideas and concepts in a modular way, and at multiple levels.); Sharing of statistical strength for multi-task learning; Computational complexity [13]. In fact, it was found recently that the features learnt in deep architectures resemble those observed in the

ISSN: 2005-4254 IJSIP

areas V1 and V2 of visual cortex [14], and that they become more and more in variant to factors of variation in higher layers. Learning a hierarchy of features increases the ease and practicality of developing representations that are at once tailored to specific tasks, yet is able to borrow statistical strength from other related tasks. Finally, learning the feature representation can lead to higher-level (more abstract, more general) feature that are more robust to unanticipated sources of variance extant in real data.

Deep learning uses a lot of data that is often easily obtained even in massive quantities, and that thus can provide a large number of "bits" of information for algorithms to try to learn from. Thus, we believe that if good deep learning algorithms can be developed, they hold the potential to make machine learning significantly more effective for many problems. In this paper, we present an algorithm for image restoration task that combines sparse coding and deep networks pre-trained with denoising auto-encoder (DAE) defined by this model, and show that by training on large image databases we are able to outperform the current state-of-the-art image denoising methods. Our algorithms will first learn a large basis functions, and then reconstruct any new input image using a weighted combination of a few of these basis functions. The weights of these basis functions then give a slightly higher-level and more succinct representation of the input; this representation can then be used in image restoration task.

2. The Model

The basic framework for our models is the auto-encoder (AE)[12].An basic auto encoder is comprised of an encoder function $h(.)$ maps an input$x \in R^d$ to some hidden representation $h(x) \in R^{d_h}$, anda decoder $g(.)$ maps this hidden representation back to areconstructed version of$x$, such that$g(h(x)) \approx x$. The parameters of the auto encoders are learned to minimize their construction error, measured by some loss $\ell\left(x, g\left(h(x)\right)\right)$. Examples ofreconstruction error include the cross-entropy loss (for binary x), or like here, squared error(for real-valued$x$)

$$L(x, g(h(x))) = L\_2(x, g(h(x))) = |(|x - g(h(x))|)|^2.$$

Denoising Auto encoders (DAE) [9] incorporate a slight modification to this setup and corrupt the inputs before mapping them into the hidden representation. They are trained to reconstruct (or denoise) the original input x from its corruptedversion $\tilde{x}$ by minimizing

$$L\left(x, g\left(h(\tilde{x})\right)\right) = L_2\left(x, g\left(h(\tilde{x})\right)\right) = \left|\left|x - g\left(h(\tilde{x})\right)\right|\right|^2.$$

Typical choices of corruption include additive white Gaussian noise (AWG) or binary masking noise. In this work, we use the former with standard deviation σ.This is a rational choice for natural images captured by a digital camera. The DAE architecture is shown in Figure 1.

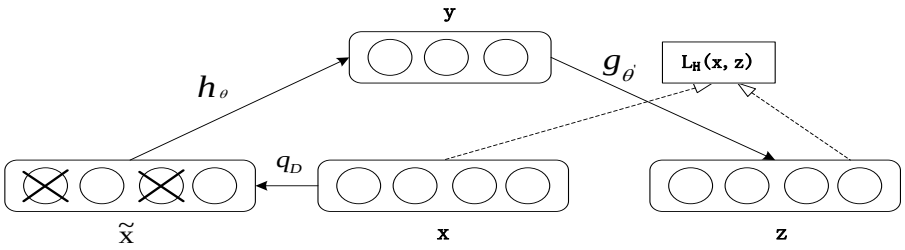


Figure 1. The DAE Architecture. An Example $x$is Stochastically Corrupted (via q_d) to $\tilde{x}$

The auto encoder then maps it to y (via encoder h_θ) and attempts to reconstruct x via decoder $g_{\theta'}$, producing reconstruction **z**. Reconstruction error is measured by loss $L_H(\boldsymbol{x}, \boldsymbol{z})$.

To use DAE for deep learning, we follow the general greedy layer-wise procedure [16] and pre-train each layer of a deep neural network as a DAE. During the greedy pre-training phase, when training the ithlayer, each input is mapped to its hidden representation$h_{i-1}(x)$ and is used as a training sample to aDAE. Note that this requires the corruption of $h_{i-1}(x)$ into$\widetilde{h_{i-1}(x)}$, forcing the hidden units to represent the leading regularities in the data. A layer is pre-trained for a fixed number of updates, after which the new representation is used as input for the next layer. Greedy pre-training then move son to the next hidden layer. The complete procedure for learning and stacking several layers of denoising auto encoders is shown in Figure 2. In the following experiments section, we follow this approach to initialize the weights and subsequently fine tune the network with the stochastic back propagation.

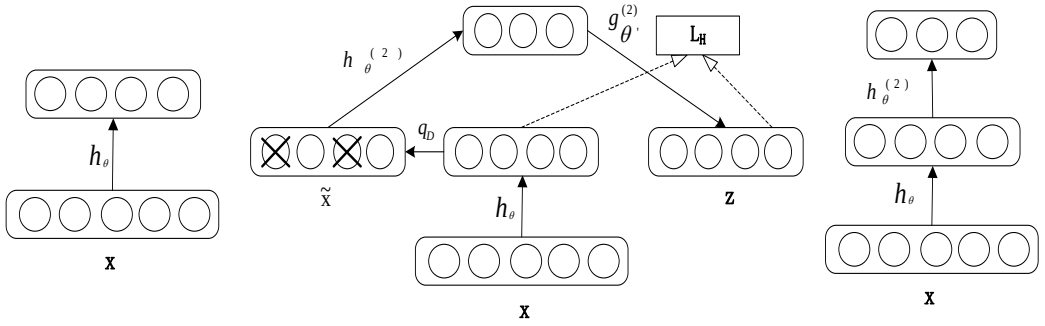


Figure 2. Stacking Denoising Auto Encoders

After training a first level denoising auto encoder (see Figure 1) its learnt encoding function h_θ is used on clean input (left). The resultingrepresentation is used to train a second level denoising auto encoder (middle) to learn a second level encoding function $h_\theta^{(2)}$. From there, the procedure can be repeated (right).

3. Learning to Denoise

We performed all our experiments on grey-scale images, but there is no difficulty in generalizing to colored images. Image denoising aim to map a noisy image to a cleaner version. However, the complexity of a mapping from images to images is large, so in practice we chop the image into possibly overlapping patches and learn a mapping from a noisy patch

to a clean patch. To denoise a given image, all image patches are denoised separately by that map. The denoised image patches are then combined into a denoised image.

To evaluate the model we use a set of 60, 000 images from the CIFAR-bw data set. Our system performs the following steps to feature extraction and image restoration:

a. Extract random patches from training images.
b. Apply a pre-processing stage to the patches.
c. Learn a feature-mapping using stacked DAE learning algorithm.
d. Train a image restoration algorithm.

We will now describe the components of this pipe line and its parameters in more detail.

3.1. Feature Learning

We have tested our approach on a benchmark image sets, namely: CIFAR-bw: a gray-scale version of the CIFAR-100[17].The CIFAR-100 is labeled subsets of the 80 million tiny images dataset. They were collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The Sample images are shown in Figure 3.

Figure 3. Sample Images from the CIFAR-bw Image Sets

Next, we get corrupted images. Assuming $\tilde{x}$ is the observed noisy image and x is the original noise free image, we can formulate the image corruption process as:

$$\tilde{x} = q(x). \quad (1)$$

Where $q: R^n \rightarrow R^n$is an arbitrary stochastic corrupting process that corrupts the input? For most of our experiments, we used AWG noise with$\sigma = 25$. However, we also show results for other noise levels.

Finally, as mentioned above, the system begins by extracting random sub-patches from input images. Each patch has dimension $N = n \times n$. Each$n \times n$ patch can be represented as a vector in R^N ofpixel intensity values. We thenconstruct a dataset of m randomly sampled patches, $X = \{x^{(1)}, x^{(2)} \ldots, x^{(m)}\}$ where $x^{(i)} \in R^N$. Given thisdataset, we apply the pre-processing and unsupervisedlearning steps.

3.1.1. Pre-processing

Data preprocessing plays a very important in many deep learning algorithms. In practice, many methods work best after the data has been normalized. In this work, we assume that every patch is normalized by simple re scaling (dividing the patch by 255), subtracting the mean and dividing by the standard deviation of its elements. For visual data, this corresponds to local brightness and contrast normalization.

3.1.2. Feature Extraction

The basic building block of our framework is a one-layer DAE. The DAE tries to learn a function$h_{w,b}(\tilde{x}) \approx x$, which minimizes the squared reconstruction loss:

$$J(W,b) = \frac{1}{m}\sum_{i=1}^{m} \| h_{w,b}\left(\widetilde{x^{(i)}}\right) - x^{(i)} \|^2 + \frac{\lambda}{2}\sum W + \beta KL \qquad (2)$$

Where

$$KL = KL(\rho \| \varrho_j) = \rho \log\frac{\rho}{\varrho_j} + (1-\rho)\log\frac{1-\rho}{1-\varrho_j},\ \varrho_j = \frac{1}{m}\sum_{i=1}^{m} h_{w,b}\left(\widetilde{x^{(i)}}\right). \qquad (3)$$

The first term in the definition of $J(W,b)$ is an average sum-of-squares error term. The second term is a regularization term (also called a weight decay term) that tends to decrease the magnitude of the weights, and helps prevent over fitting. The third term is a sparsity penalty term then enforce the average activation of hidden is a small value close to zero.λ and β controls the weights of the penalty term. We choose to minimize the squared error since it is monotonically related to the PSNR, which is the most commonly, used measure of image quality. Thus minimizing the squared error will maximize PSNR values.

One-layer DAE is a computational unit that takes as input $\tilde{x}$, and outputs

$$h(\tilde{x}) = f(wx + b) \qquad (4),$$

Where $f: R \to R$is activation function. In this work, we will choose $f(.)$to be the sigmoid function:$f(z) = \frac{1}{1+\exp(-z)}$.

We evaluate different hidden layers, and find that it is not always beneficial to add hidden layers. A possible explanation is that SDAE with more hidden layers become more difficult to learn. Indeed, each hidden layer adds non-linearities to the model. It is therefore possible that the error landscape is complex and that stochastic gradient descent gets stuck in a poor local optimum from which it is difficult to escape. In the meantime, we try different patch sizes and find that higher noise level generally requires larger patch size.

3.2. Image Restoration

We use standard testing images that have been used to evaluate other denoising algorithms as the testing set. To denoise images, we decompose a given noisy image into overlapping patches. We then normalize the patches (see Section 3.1.1), denoise each patch separately and perform the inverse normalization on the denoised patches. The denoised image is obtained by placing the denoised patches at the locations of their noisy counterparts, then averaging on the overlapping regions.

After denoising an image, we would like to know: How good is the denoising result? A possible solution to this problem would be to rely on human evaluation of the image quality. However, this solution is too inconvenient for many applications. Hence, one is interested in automatic image quality assessment and in particular in objective image quality metrics that correlate with subjective image quality.

There are many image quality metrics, include peak signal-to-noise ration (PSNR [18]), structural similarity index (SSIM [19]), information-content weighted PSNR (IW-PSNR [20]), the information fidelity criterion (IFC[21]), DIIVINE[22], LBIQ[23], BIQI[24], *etc.*,

We employ PSNR which is the most commonly used metric to quantify denoising results. The PSNR is computed by

$$\text{PSNR} = 20\log_{10}(\frac{255}{\sigma_e^2}),$$

Where σ_e^2 is the mean squared error.

4. Experiments

In this section, we demonstrate the results achieved by applying the above methods on several test images. Before we present denoising result, we first show visualizations of the learned feature representations. The bases learned by SDAE are shown in Figure 4 for 8 pixel receptive fields.

We have compared our method to two well-known and widely-available denoising algorithms: KSVD [3] (a dictionary-based method) and BM3D [6] (a block matching procedure).

Table 1 compares our method against KSVD and BM3D on the test set of 4 standard test images. The result from left to right is KSVD, BM3D and our method.

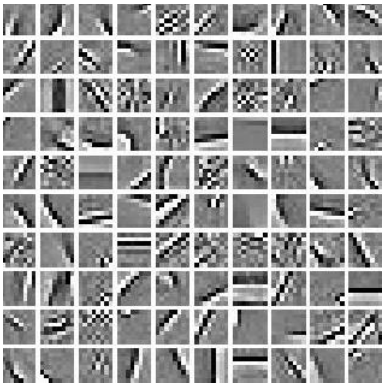


Figure 4. Randomly Selected Bases Trained on CIFAR-bw Images Set

Table 1. Comparison of the Denoising Performance

σ/PSNT	Lena(512×512)			Barbara(512×512)			Boats(512×512)			House(256×256)		
10/	35.46	**35.92**	35.82	34.40	**34.96**	34.01	33.65	**33.89**	33.81	35.94	**36.71**	35.98
25/	31.32	32.22	**32.23**	29.60	**30.99**	29.69	29.28	**30.03**	29.95	32.15	**32.95**	32.58
50/	27.43	28.99	**29.28**	25.22	**27.21**	25.29	25.90	26.72	**26.97**	27.44	**29.71**	29.62
75/	24.87	27.16	**27.63**	22.65	**25.10**	23.45	23.59	25.04	**25.37**	24.53	27.46	**27.84**

A visual comparison is shown in Figure 5. We set two hidden layers of size 500, AWG noise with $\sigma = 25$, the other hyper-parameters are:$\lambda = 10^{-4}$,$\beta = 0.035$,$\rho = 0.01$.

The experimental results show that BM3D perform better than other methods on average. Analyzing the outcomes of those experiments, we conclude than BM3D based on knowledge about the image to be denoised perform well on images with regular structure (*e.g.*, image "Barbara"), where as our methods based on knowledge about all images perform well on images with complex structures(*e.g.*, image "Lena")or high noise levels.

5. Conclusion and Future Work

In this paper, we have described an algorithm for image denoising task defined by the deep learning framework. We have compared the results achieved by our approach against other algorithms, and show that by training on large image databases we are able to outperform the current state-of-the-art image denoising methods.

In our future work, we would like to explore the possibility of adapting the proposed approach to various other applications such as denoising and in painting of text and audio. It is also meaningful to investigate into the effects of different hyper parameter settings on the learned features.

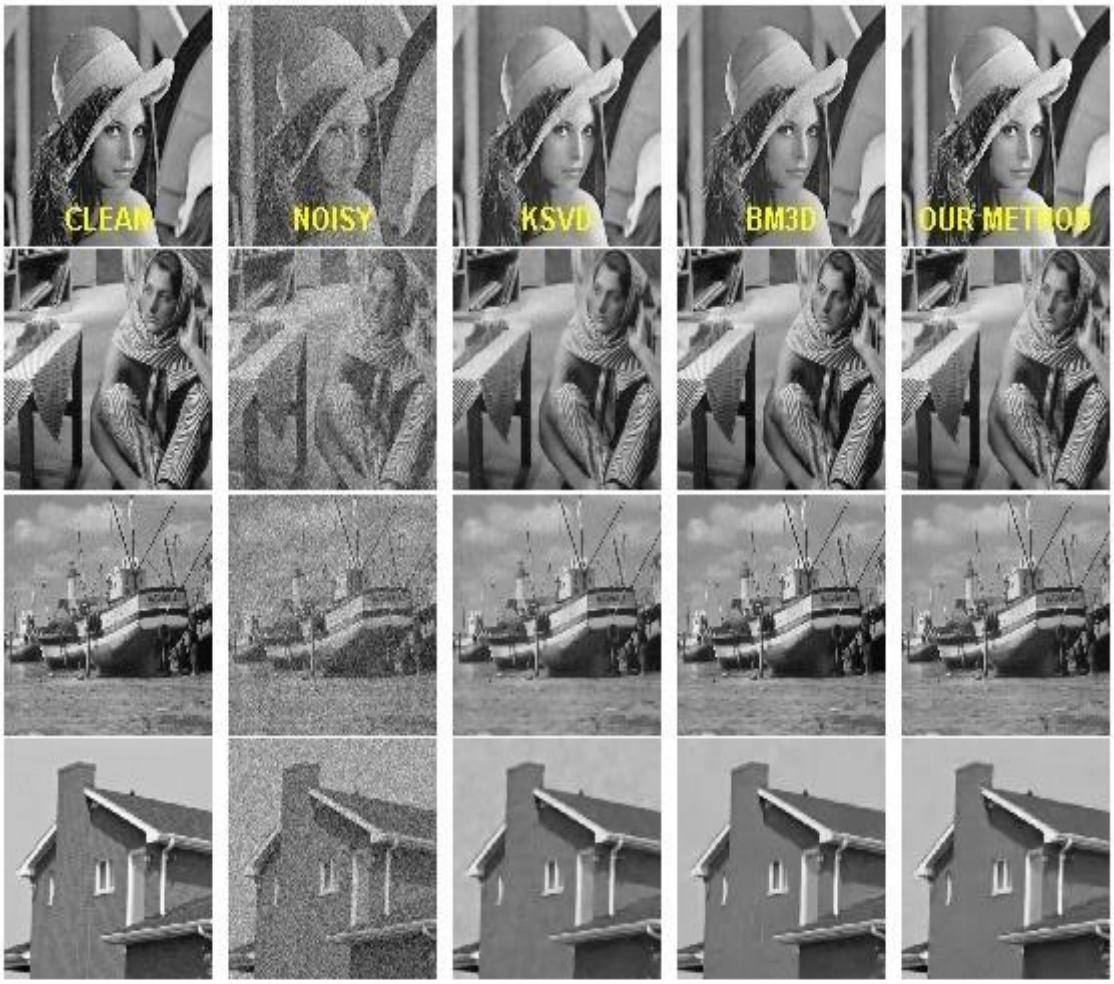
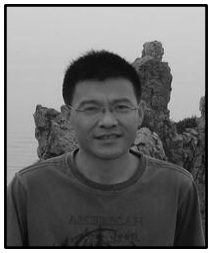


Figure 5. Visual Comparison of Denoising Results

References

[1] F. Luisier, T. Blu and M. Unser, "A new SURE approach to image denoising: Inter scale ortho normal wavelet thres holding", Image Processing, IEEE Transactions on, vol. 16, no. 3, **(2007),** pp. 593-606.
[2] J. Portilla, V. Strela, M.J. Wainwright and E.P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain", Image Processing, IEEE Transactions on, vol. 12, no. 11, **(2003),** pp. 1338–1351.
[3] M. Aharon, M. Elad and A. Bruckstein, "K-svd: An algorithm for designing over complete dictionaries for sparse representation", IEEE Transactions on Signal Processing (TIP), vol. 54, no. 11, **(2006),** pp. 4311-4322.
[4] J. Mairal, F. Bach, J. Ponce, G. Sapiro and A. Zisserman, "Non-local sparse models for image restoration", In International Conference on Computer Vision (ICCV). IEEE, **(2010).**
[5] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole imagerestoration", In International Conference on Computer Vision (ICCV). IEEE, **(2011).**
[6] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative _ltering", IEEE Transactions on Image Processing (TIP), vol. 16, no. 8, **(2007),** pp. 2080-2095.
[7] Y. Tang, R. Salakhutdinov, G. Hinton and R. Boltzmann, "Machines for recognition and denoising", Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, **(2012),** pp. 2264-2271.
[8] P. Vincent, H. Larochelle, Y. Bengio, *et al.*, "Extracting and composing robust features with denoising auto encoders", Proceedings of the 25th international conference on Machine learning. ACM, **(2008),** pp. 1096-1103.
[9] P. Vincent, H. Larochelle, I. Lajoie, *et al.*, "Stacked denoising auto encoders: Learning useful representations in a deep network with a local denoising criterion", The Journal of Machine Learning Research, vol. 11, **(2010),** pp. 3371-3408.
[10] H. C. Burger, C. J. Schuler and S. Harmeling, "Image denoising: Can plain neural networks compete with bm3d?", IEEEConf. Comput. Vision and Pattern Recognition, **(2012),** pp. 2392–2399.
[11] H. C. Burger, C. J. Schuler and S. Harmeling, "Image denoising with multi-layer perceptrons, part 1: comparison with existing algorithms and with bounds", arXiv, **(2012),** pp. 1211-1544.
[12] Y. Bengio, "Learning deep architectures for A.I.", Foundations and Trends in Machine Learning, vol. 2, no. 1, **(2009),** pp. 1-127.
[13] Y. Bengio and A. Courville, "Deep Learning of Representations", Handbook on Neural Information Processing, Springer Berlin Heidelberg, **(2013),** pp. 1-28.
[14] H. Lee, C. Ekanadham and A. Ng, "Sparse deep belief net model for visual area V2. In NIPS'07, Cambridge, MIT Press, MA, **(2008),** pp. 873–880.
[15] R. Raina, A. Battle, H. Lee, *et al.*, "Self-taught learning: transfer learning from unlabeled data", Proceedings of the 24th international conference on Machine learning. ACM, **(2007),** pp. 759-766
[16] Y. Bengio, P. Lamblin, D. Popovici, *et al.*, "Greedy layer-wise training of deep networks", Advances in neural information processing systems, vol. 19, **(2007),** pp. 153.
[17] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images", Master's thesis, Department of Computer Science, University of Toronto, **(2009).**
[18] Q. Huynh-Thu, M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment", Electronics letters, vol. 44, no. 13, **(2008),** pp. 800-801.
[19] Z. Wang, A. C. Bovik, H. R. Sheikh, *et al.*, "Image quality assessment: From error visibility to structural similarity", Image Processing, IEEE Transactions on, vol. 13, no.4, **(2004),** pp. 600-612.
[20] Z. Wang and Q. Li, "Information content weighting for perceptual image quality assessment", Image Processing, IEEE Transactions on, vol. 20, no. 5, **(2011),** pp. 1185-1198.
[21] H. R. Sheikh, A. C. Bovik, G. De Veciana, "An information fidelity criterion for image quality assessment using natural scene statistics", Image Processing, IEEE Transactions on, vol. 14, no. 12, **(2005),** pp. 2117-2128.
[22] A. K. Moorthy and A. C. Bovik, "Blind image quality assessment: From natural scene statistics to perceptual quality", Image Processing, IEEE Transactions on, vol. 20, no. 12, **(2011),** pp. 3350-3364.
[23] H. Tang, N. Joshi and A. Kapoor, "Learning a blind measure of perceptual image quality Computer Vision and Pattern Recognition (CVPR)", IEEE Conference on. IEEE, **(2011),** pp. 305-312.
[24] A. K. Moorthy and A. C. Bovik, "A two-step framework for constructing blind image quality indices", Signal Processing Letters, IEEE, vol. 17, no. 5, **(2010),** pp. 513-516.

Authors



HuiMing Li, received the M. Eng in Computer science and Application from Liaoning Normal University in 2006. He current research interests are in the fields of machine learning and data mining.

