

Image Blur Classification and Parameter Identification using Two-stage Deep Belief Networks

Ruomei Yan
r.yan@sheffield.ac.uk

Ling Shao
ling.shao@sheffield.ac.uk

Department of Electronic and Electrical
Engineering,
The University of Sheffield,
Sheffield, UK

Abstract

Image blur kernel classification and parameter estimation are critical for blind image deblurring. Current dominant approaches use handcrafted blur features that are optimized for a certain type of blur, which is not applicable in real blind deconvolution application when the Point Spread Function (PSF) of the blur is unknown. In this paper, a Two-stage system using Deep Belief Networks (TDBN) is proposed to first classify the blur type and then identify its parameters. To the best of our knowledge, this is the first time that Deep Belief Network (DBN) has been applied to the problem of blur analysis. In the blur type classification, our method attempts to identify the blur type from mixed input of various blurs with different parameters, rather than blur estimation based on the assumption of a single blur type in current methodology. To this aim, a semi-supervised DBN is trained to project the input samples in a discriminative feature space, and then classify those features. Moreover, in the parameter identification, the proposed edge detection on logarithm spectrum helps DBN to identify the blur parameters with very high accuracy. Experiments demonstrate the effectiveness of the proposed methods with better results compared to the state-of-the-art on the Berkeley segmentation dataset and the Pascal VOC 2007 dataset.

1 Introduction

Image blur is a major source of image degradations, although it can sometimes be required for artistic purposes. Various sources can cause image blur, such as the atmospheric turbulence (Gaussian blur), camera relative motion during exposure (motion blur), and lens aberrations (out-of-focus blur) [16]. Assuming that we consider spatially invariant blurs, the formation processes of these blurs can be described by global convolution models, which will be explained in Sec.2.1.

The restoration of blurred photographs, image deblurring, is the process of inferring latent sharp images with inadequate information of the degradation model. It can be categorized into *blind* and *non-blind*. Non-blind deblurring requires the prior knowledge of the blur kernel and its parameters, while in blind deblurring we assume that the blurring operator is unknown. In most situations of practical interest the Point Spread Function (PSF) is not acquired, so the application range of non-blind deblurring is much narrower than the

blind deblurring [10]. In blind image deblurring, there are two main classes: *multi-image* [9, 11, 12] and *single-image* deblurring. In the real application, a single blurred image is usually the case we need to deal with. For instance, Single-channel blind deconvolution within Bayesian framework is considered in [13]. The parameter for Gaussian blur is determined efficiently by wavelet decomposition in [17]. Restoring images degraded by motion blur is discussed in [26]. Similarly, one popular approach is the application of radon transform, which can estimate the blur kernel by analyzing the edges in the image [8]. Other methods have also been tried for motion blur, such as cepstral method, and steerable filters [14].

While most previous work focuses on image deblurring, not as much research has been done on *blur classification*, which is more practical because the type of blurs is usually unknown in photographs. Based on the descriptor of blurs, there are a few blur classification method without doing image deblurring. One of the state-of-the-art method is a Bayes Classifier based using blur features, for instance, local autocorrelation congruency [19]. Another similar method has been proposed by Su *et al.* [29] based on the alpha channel feature, which has different circularity of the blur extension. Though both of them managed to detect local blurs in the realistic image, their methods are based on handcrafted features.

Although previous blur classification methods can perform well with handcrafted features, their performance is still limited due to the diversity of natural images. Recently, many researchers have moved their attention from the heuristic prior to the learned deep architecture. The deep hierarchical neural network roughly mimics the nature of the mammalian visual cortex, which has been applied in many vision tasks, such as object recognition, image classification, and even image analysis. In Jain's denoising work [15], they have shown the potential of using Convolutional Neural Network (CNN) for denoising images corrupted by Gaussian noise. In such an architecture, the learned weights and biases in the deep convolutional neural network are obtained through the training on sufficient amount of natural images. For testing stage, these parameters in the network act like 'prior' information for the degraded images, which end up with better results compared to the top local denoising approaches. Another example is the blur extent metric developed by the multifeature classifier based on Neural Networks (NN) [9]. It has proved that the combined learned feature works better than individual handcrafted feature mostly.

Inspired by the practical blur type classification in [19, 29] and the merits of learned descriptors in [9, 13], we intend to design another patch-based blur type classification and parameters identification method to better solve the realistic blur analysis problem. Deep Belief Network (DBN) is chosen for accomplishing the feature extraction and final classification in this system. A two-stage framework is proposed: first, for the input image patches with different blurs, the DBN is used for identifying the blur type; second, different samples with the same blur type will be sent to the corresponding DBN blocks for further parameter estimation. The DBN is trained in a semi-supervised way: the unsupervised training of the DBN is done by a greedy layer-wise pre-training before the supervised backpropagation for the fine-tuning. The unsupervised process helps the feature learning, and the backpropagation helps to construct the discriminative information.

In a word, our contributions are threefold:

- To the best of our knowledge, this is the first time that deep belief network has been applied to the problem of blur analysis.
- A discriminative feature, derived from edge extraction on Fourier Transform coefficients, has been proposed to preprocess blurred images before they are fed into the DBN.

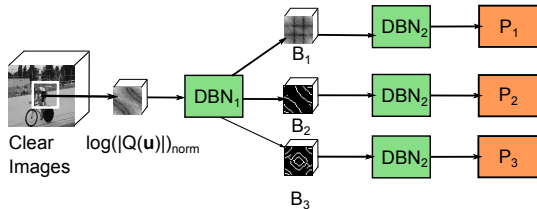


Figure 1: The TDBN architecture: DBN_1 is the first stage for blur type classification, which has 3 output labels. DBN_2 is the blur PSF parameter identification, which has different output labels for each blur type. P_1 , P_2 , and P_3 are the estimated parameter labels, which can be seen in Sec. 3.3.

- A two-stage framework is proposed to estimate the blur type and parameters for any given image degraded by spatially invariant blur of an unknown type.

2 Two-stage Deep Belief Networks

In this section, we describe the proposed two-stage deep belief networks, for blur classification and parameter identification. We explain the problem formulation, the proposed blur features, and the training of DBNs in Sec. 2.1, Sec. 2.2, and Sec. 2.3, respectively. The whole TDBN process is described in Sec. 2.4.

2.1 Problem Formulation

The image blurring can be modeled as the following degradation process from the high exposed image to the observed image [24]:

$$g(\mathbf{x}) = q(\mathbf{x}) * f(\mathbf{x}) + n(\mathbf{x}) \quad (1)$$

where $\mathbf{x} = \{x_1, x_2\}$ denotes the coordinates of an image pixel, g represents the blurred image, f is the intensity of the original high quality image, q denotes the PSF of a certain blur type, $*$ indicates the convolution, and n is the additive noise.

In blind image deconvolution, it is very difficult to recover the PSF from a single blurred image due to the loss of information during blurring [24]. Our goal is to classify the blurred patches into their corresponding degradation functions and parameters. Several blurring functions are considered in this paper.

In many applications, such as satellite imaging, Gaussian blur can be used to model the PSF of the atmospheric turbulence:

$$q(\mathbf{x}, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x_1^2 + x_2^2}{2\sigma^2}\right), \quad \mathbf{x} \in R \quad (2)$$

where σ is the blur radius to be estimated, and R is the region of support. R is usually set as $[-3\sigma, 3\sigma]$, because it contains 99.7% of the energy in a Gaussian function [6].

Another blur is caused by linear motion of the camera, which is called motion blur [15]:

$$q(\mathbf{x}) = \begin{cases} \frac{1}{M}, & \text{if } f(x_1, x_2) \begin{pmatrix} \sin(\omega) \\ \cos(\omega) \end{pmatrix} = 0 \quad \text{and} \quad x_1^2 + x_2^2 \leq M^2/4 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where M describes the length of motion in pixels and ω is the motion direction with its angle to the x axis. These two parameters are what we need to estimate in our system.

The third blur is the out-of-focus blur, which can be modeled as a cylinder function:

$$q(\mathbf{x}) = \begin{cases} \frac{1}{\pi R^2}, & \sqrt{x_1^2 + x_2^2} \leq R \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where the blur radius R is proportional to the extent of defocusing.

In the blur classification method of [15], a motion blur descriptor, local autocorrelation congruency, is used as a feature for the Bayes classifier to discriminate motion blur from defocus blur because the descriptor is strongly related to the shape and value of the PSF. Later, Su *et al.* [16] have presented better handcrafted features for blur classification, which gives better results without any training. Though both methods generate good results on identifying motion blur and out-of-focus blur, the features they used are both limited to a single or several blur kernels. In this paper, we attempt to find a general feature extractor for common blur kernels with various parameters, which is closer to realistic application scenarios. Therefore, enlightened by the previous success of applying deep belief networks to discriminative learning [5], we consider to use the DBN as our feature extractor.

When designing the DBN, it is natural to use observed blurred patches as training and testing samples. However, their characteristics are not as obvious as their frequency coefficients [5]. Hence, the logarithmic power spectra are adopted as input features for the DBN, since the PSF in the frequency domain manifests different characteristics for different blur kernels. Bengio *et al.* [3] have pointed out that the scaling continuous-valued input to $(0, 1)$ worked well for pixel gray levels, but it is not necessarily appropriate for other kinds of input data. From Eq. 1 one can see that the noise might interfere the inference in the DBN [3], so preprocessing steps are necessary for preparing our training samples. In this paper, we use an edge detector to obtain binary input values for the DBN training, which has been proved to benefit the blur analysis task.

We propose a two-stage classification system to both classify the blur kernel and identify the blur parameters. These two stages have a similar network architecture but different input layers. The first stage is an initial classification of the blur type, and the second stage is to further identify the blur parameters within samples with the same label from the results of the first stage. Since the variation between blur parameters of the same blur type is not as great as that between different blur types, more discriminative features have been designed for the second stage, which yields much better results than combining the two stages into one in our experiments.

2.2 Blur features

2.2.1 Features for motion and defocus blurs

If we apply the Fourier Transform (FT) to both sides of Eq. (1), we can obtain:

$$G(\mathbf{u}) = Q(\mathbf{u})F(\mathbf{u}) + N(\mathbf{u}) \quad (5)$$

where $\mathbf{u} = \{u_1, u_2\}$. For the out-of-focus blur, $Q(\mathbf{u}) = \frac{J_1(\pi Rr)}{\pi Rr}$, $r = \sqrt{u_1^2 + u_2^2}$.

J_1 is the first-order Bessel function of the first kind and the amplitude is characterized by almost-periodic circles of radius R along which the Fourier magnitude takes value zero.¹

For the motion blur, the FT of the PSF is a *sinc* function: $Q(\mathbf{u}) = \frac{\sin(\pi M\omega)}{\pi M\omega}$, $\omega = \pm \frac{1}{M}, \pm \frac{2}{M}, \dots$

In order to know the PSF $Q(\mathbf{u})$, we attempt to identify type and parameters of Q from the observation image $G(\mathbf{u})$. Therefore, the normalized logarithm of Q can be used in our implementation:

$$\log(|Q(\mathbf{u})|)_{norm} = \frac{\log(|Q(\mathbf{u})|) - \log(|Q_{min}|)}{\log(|Q_{max}|) - \log(|Q_{min}|)} \quad (6)$$

where Q represents $Q(\mathbf{u})$, $Q_{max} = \max_{\mathbf{u}}(Q(\mathbf{u}))$, and $Q_{min} = \min_{\mathbf{u}}(Q(\mathbf{u}))$.

As shown in Fig. 2, the patterns in these images ($\log(|Q(\mathbf{u})|)_{norm}$) can represent the motion blur or the defocus blur intuitively. Hence, no extra preprocessing needs to be done for the blur type classification. However, defocus blurs with different radii are easy to be confused, which also has been proved in our experiments. Therefore, for blur parameter identification, an edge detection step is proposed here.

Since the highest intensities concentrate around the center of the spectrum and decrease towards its borders, the binarization threshold has to be adapted for each individual pixel, which is computationally prohibitive. If a classic edge detector is applied directly, redundant edges would interfere with the pattern we need for the DBN learning. Many improved edge detectors have been explored to solve this issue, however, most of them do not apply to the logarithmic power spectra data, which cause even worse performance [2, 22]. For instance, Bao *et al.* [2] proposed to improve the Canny detector by the scale multiplication, which indeed enhances the localization of the Canny detector. However, this method does not generate good edges on our images.

We solve this issue by applying the Canny detector first, and then using a heuristic method to refine the detected edges. Due to the fact that the useful edges are isolated near zero-crossings, we need to refine the detection results from the logarithmic power spectrum. The Canny edge detector is applied to form an initial edge map. Then, we design several steps to select the most useful edges: 1) For both of the blur types, we select isolated edges. Assuming the isolated region has the radius d , those edges, in the orthogonal direction of the current edge within radius d , will be discarded [8]. 2) For the motion blur, we abandon short and very curvy edges. We consider the orientations $\theta = [0, \pi]$ of the candidate edges within radius d are considered. The criterion proposed by Watson [8] is utilized for estimating their alignment [8].

¹<http://www.clear.rice.edu/elec431/projects95/lords/elec431.html>

2.2.2 Features for the Gaussian blur

For the Gaussian blur, the Fourier transform of the PSF is still a Gaussian function, and there is no significant pattern change in the frequency domain. From Eq.(2), we can see that the Gaussian kernel serves as a low pass filter. When the sigma of this filter is larger, more 'high frequency information' will pass in. However, from our observation, when the σ is larger than 2, the pattern on the logarithmic spectrum image barely changes. In the experiment section, we show that edge detection is not suitable in this case.

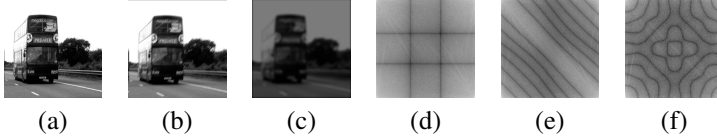


Figure 2: The blur images and their logarithmic spectra. (a) Image with Gaussian blur. (b) Image with motion blur. (c) Image with out-of-focus blur. (d) Logarithmic spectrum of Gaussian blur ($\sigma = 2$). (e) Logarithmic spectrum of motion blur ($M = 9, \omega = 45$). (f) Logarithmic spectrum of out-of-focus blur ($R = 30$).

2.3 The Training Process of Deep Belief Networks

Deep belief nets are used as a generative model for feature learning in a lot of previous work [11]. In this paper, we first construct the DBN by unsupervised greedy layer-wise training to extract features in the form of hidden layers and then apply a fine-tuning for discriminative weights in a supervised way.

The training process of an individual DBN is as follows:

1. The input layer is trained in the first Restricted Boltzmann Machine (RBM) as the visible layer. Then, a representation of the input blurred sample is obtained for further hidden layers. This representation is chosen to be the mean activations in our experiments as $p(h^{k+1}) = 1|h^k, k = 0, 1, \dots, P$, where P is the number of all the hidden layers.
2. The next layer is trained as an RBM by greedy layer-wise information reconstruction. The training process of RBM is to update weights between two adjacent layers and the biases of each layer. In our scheme, Contrastive Divergence (CD) [12] is applied.
3. Repeat the first and second steps until the parameters in all the layers (visible and all hidden layers) are learned.
4. In the supervised learning part, the labels are used for training the DBN to have discriminant ability using backpropagation. Then, the goal for the optimization process is to minimize the backpropagation error derivatives: $\phi^* = \operatorname{argmin}_{\phi} [-\sum_p \mathbf{y}_p \log \hat{\mathbf{y}}_p]$, where \mathbf{y}_p and $\hat{\mathbf{y}}_p$ are the estimated label and the correct label. The conjugate gradient descent is used for this optimization.

2.4 Forming the TDBN

The TDBN is formed by two-stage DBN learning (Fig. 1). First, the identification of blur patterns is carried out in the first stage by using the logarithmic spectra of the input blurred patches. The output of this stage is 3 labels: the Gaussian blur, the motion blur and the defocus blur. With the label information, the classified blur vectors will be used in the second stage for blur parameter identification. At this stage, motion blur and defocus blur will be further preprocessed by the edge detector (Sec. 2.2) before the training but Gaussian blur vectors remain the same. The output of this stage is various labels for individual DBNs as shown in Sec. 3.3.

3 Experiments

3.1 Experimental setup

Training datasets: *The Oxford image classification dataset*², and *the Caltech 101 dataset* are chosen to be our training sets. We randomly selected 4000 images from each of them.

The size of the training samples ranges from 32×32 to 128×128 pixels, which are cropped from the original image. By empirical evaluation, the best results occur when the patch size is 32×32 . The size of the training set is 12000 (randomly selected from those cropped images). In those 12000 training samples, 4000 of them are degraded by Gaussian PSF, 4000 of them are degraded by the PSF of motion blur, and the rest are degraded by the defocus PSF.

Testing datasets: *Berkeley segmentation dataset* (200 images) has been used for our testing stage, which has been applied to the denoising algorithms [24, 28]. *Pascal VOC 2007*: 500 images are randomly selected from this dataset [20].

2000 testing samples are chosen from each of them according to the same procedure as the training set. The numbers of the three types of blurred patches are random in the testing set.

Blur features: The Canny detector is applied to the logarithmic power spectrum of image patches with automatic low and high thresholds. Afterwards, the isolated edges are selected with the radius of 3 pixels according to the suggestions from [8].

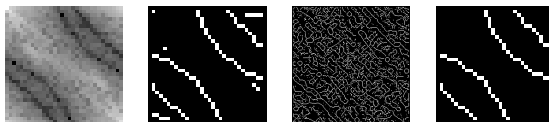


Figure 3: Comparison of the three edge detection methods applied to a training sample. From left to right: (1) the logarithmic power spectrum of a patch; (2) the edge detected by Canny detector (automatic thresholds); (3) the edge detected by the improved Canny detector using scale multiplication; (4) the edge detected by our method

DBN Training: For parameters of the DBN learning process, the basic learning rate and momentum in the model are set according to the previous work [3]. In the unsupervised greedy learning stage, the number of epochs is fixed at 50 and the learning rate is 0.1. The

²<http://www.robots.ox.ac.uk/vgg/share/practical-image-classification.htm>

Method	Features	CR1	CR2
Liu <i>et al.</i> 's method [19]	Handcrafted	79.3%	80.5%
Su <i>et al.</i> 's method [29]		81.6%	83.1%
SVM on our features [10]		78.2%	80.8%
NN [23]	Learned	92.5%	91.7%
CNN [25]		95.3%	96.8%
Proposed		99.7%	98.2%

Table 1: Comparison of obtained average results on the two testing datasets with the state-of-the-art. CR1 is the Berkeley dataset, and CR2 is the Pascal dataset.

initial momentum is 0.5, and it changes to 0.9 after five epochs. Our supervised fine-tuning process always converges earlier than epoch 30.

3.2 Image blur type classification

In our implementation, the input visible layer has 1024 nodes, and the output layer has 3 labels (Gaussian kernel, motion kernel, and defocus kernel). Therefore the whole architecture is: $1024 \rightarrow 500 \rightarrow 30 \rightarrow 10 \rightarrow 3$. These node numbers in each hidden layer are selected empirically.

On the one hand, we compare our method with the previous blur type classification methods based on handcrafted features: [19] and [29]. Their original frameworks contain a blur detection stage, and the blur type classification is applied afterwards. However, in our algorithm, the image blurs are simulated by convolving the high quality patches with various PSFs. In our comparison, [19] has been trained and tested with the same datasets we used, while [29] has been tested with the same testing set we used.

On the other hand, NN [23], CNN [25] and Support Vector Machine (SVM) have been chosen for the classifier comparison. The same blur feature vectors are used for NN and CNN. The SVM-based classifier was implemented following the usual technique: several binary SVM classifiers are combined to the multi-classifier [10].

The classification rate is used for evaluating the performance:

$$CR = 100 \frac{N_c}{N_a} (\%) \quad (7)$$

where N_c is the number of the correct classified samples, and N_a is the number of the total samples.

We can observe from Table 1 that algorithms based on learned features perform better than those based on handcrafted features, which suggests that learning based feature extractor is less restricted to the type of the blur we consider. Meanwhile, our method performs best among all the algorithms using automatically learned features.

3.3 Blur kernel parameter identification

In this experiment, the parameters of the blur kernels are identified. For different blur kernels, different parameters are estimated as explained in Sec. 2.1. The parameters are set as: 1) Gaussian blur has 8 labels: $\sigma = \{0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4\}$; 2) Motion blur has 8 labels: $M = \{3, 9\}$ $\omega = \{0, 45, 90, 135\}$; 3) Out-of-focus blur has 8 labels: $R =$

Method	CR11	CR12	CR13	CR21	CR22	CR23
SVM [10]	96.5%	97.2%	96.9%	95.1%	95.7%	94.9%
NN [23]	90.1%	92.6%	92.2%	90.9%	91.5%	90.6%
CNN [23]	97.9%	98.9%	98.5%	97.3%	98.1%	98.2%
DBN	97.8%	98.1%	97.9%	97.7%	97.8%	97.5%
TDBN1	99.5%	98.8%	98.4%	99.3%	98.5%	98.2%
TDBN2	99.2%	99.9%	99.4%	99.1%	99.7%	99.2%

Table 2: Comparison of obtained results on the two testing datasets with the state-of-the-art. In CRxx the first x refers to the dataset type (1 for Berkeley and 2 for Pascal) and the second x refers to the blur type (the Gaussian blur, the motion blur, and the defocus blur). DBN is the case that the mixed blur patches are classified by a single DBN. TDBN1 is the case when we use the logarithm spectrum for stage 1 and stage 2. TDBN2 is the case when we use the logarithmic spectrum for stage 1 and edge detection for stage 2.

{2, 5, 8, 11, 14, 17, 20, 23}. The architecture in each DBN is the same except for the output layer: $1024 \rightarrow 576 \rightarrow 36 \rightarrow 25 \rightarrow$ number of labels for a certain blur type.

Our method is compared to the NN, CNN, and SVM with the same input layer of the blur features. As shown in the following Table 2, our method achieves the best results among all, especially for the motion and defocus blur due to the obvious patterns they have in their logarithmic power spectra. Besides, for the Gaussian blur, we can observe that the edge detector has not benefited them, which is consistent with our previous analysis. Moreover, our proposed two-stage strategy works better than a single DBN as shown in Table 2.

4 Conclusions

In this paper, a two-stage deep belief network has been proposed for the blur type classification and parameter identification. Our training samples are generated by patches from abundant datasets, after the Fourier transform and our designed edge detection. In the training stage, deep belief networks have been applied in a semi-supervised way. That is, the whole network is trained in an unsupervised manner and afterwards the backpropagation fine-tunes the weights. In this way, a discriminative classifier has been trained. The experimental results have demonstrated the superiority of our TDBN compared to the state-of-the-art methods.

References

- [1] M. Almeida and L. Almeida. Blind and semi-blind deblurring of natural images. *IEEE Transactions on Image Processing*, 19(1):36–52, 2010.
- [2] P. Bao, L. Zhang, and X. Wu. Canny edge detection enhancement by scale multiplication. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(9):1485–1490, 2005.
- [3] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Proc. NIPS*, pages 153–160, 2006.

-
- [4] J. Cai, H. Ji, C. Liu, and Z. Shen. High-quality curvelet-based motion deblurring from an image pair. In *Proc. CVPR*, pages 1566–1573, 2009.
- [5] M. Cannon. Blind deconvolution of spatially invariant image blurs with phase. *IEEE Transactions on Acoustic Speech, and Signal Processing*, 24(1):58–63, 1976.
- [6] F. Chen and J. Ma. An empirical identification method of gaussian blur parameter for image deblurring. *IEEE Transactions on Signal Processing*, 57(7):2467–2478, 2009.
- [7] J. Chen, L. Yuan, C. Tang, and Quan L. Robust dual motion deblurring. In *Proc. CVPR*, pages 1–8, 2008.
- [8] T. Cho, S. Paris, B. Horn, and Freeman W. Blur kernel estimation using the radon transform. In *Proc. CVPR*, pages 241–248, 2011.
- [9] A. Ciancio, A. Costa, E. Silva, A. Said, R. Samadani, and P. Obrador. No-reference blur assessment of digital pictures based on multifeature classifiers. *IEEE Transactions on Image Processing*, 20(1):64–75, 2011.
- [10] K. Duan and S. Keerthi. Which is the best multiclass svm method? an empirical study. *Lecture Notes in Computer Science*, 3541(8), 2005.
- [11] G. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.
- [12] W. Hu, J. Xue, and N. Zheng. Psf estimation via gradient domain correlation. *IEEE Transactions on Image Processing*, 21(1):386–392, 2012.
- [13] V. Jain and H. Seung. Natural image denoising with convolutional networks. In *Proc. NIPS*, pages 769–776, 2008.
- [14] F. Kraemer, Y. Lin, B. McAdoo, K. Ott, J. Wang, D. Widemann, and B. Wohlberg. Blind image deconvolution: motion blur estimation. In *Technical Report, Mathematical Modeling in Industry X Workshop, Institute for Mathematics and Its Applications, University of Minnesota*, 2006.
- [15] D. Kundur and D. Hatzinakos. Blind image deconvolutions. *IEEE Signal Processing Magazine*, pages 43–63, 1996.
- [16] R. Lagendijk and J. Biemond. *Basic methods for image restoration and identification*. Academic Press, 2000.
- [17] Y. Li, S. Kang, N. Joshi, Seitz S., and Huttenlocher D. Generating sharp panoramas from motion-blurred videos. In *Proc. CVPR*, pages 2424–2431, 2010.
- [18] A. Likas and N. Galatsanos. A variational approach for bayesian blind image deconvolution. *IEEE Transactions on Signal Processing*, 52(8):2222–2233, 2004.
- [19] R. Liu, Z. Li, and Jia J. Image partial blur detection and classification. In *Proc. CVPR*, pages 23–28, 2008.
- [20] C. Williams J. Winn M. Everingham, L. Van Gool and A. Zisserman. The pascal visual object classes challenge 2007 results. <http://www.pascalnetwork.org/challenges/voc/voc2007/workshop/index.html>.

- [21] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. ICCV*, pages 416–423, 2001.
- [22] W. McIlhagga. The canny edge detector revisited. *International Journal of Computer Vision*, 91(3):251–261, 2011.
- [23] T. Mitchell. *Machine learning*, 1997.
- [24] R. Molina, J. Mateos, and A. Katsaggelos. Blind deconvolution using a variational approach to parameter, image, and blur estimation. *IEEE Transactions on Image Processing*, 15(12):3715–3727, 2006.
- [25] R. Palm. Prediction as a candidate for learning deep hierarchical models of data. *Master's thesis, Technical University of Denmark, DTU Informatics*, 2012.
- [26] I. Rekleitis. Optical flow recognition from the power of spectrum of a single blurred image. In *Proc. ICIP*, pages 791–794, 1996.
- [27] F. Rooms, W. Philips, and J. Portilla. Parametric psf estimation via sparseness maximization in the wavelet domain. In *Proc. SPIE vol. 5607, Wavelet Application in Industrial Processing II*, pages 26–33, 2004.
- [28] S. Roth and M. J. Black. Field of experts: a framework for learning image priors. In *Proc. CVPR*, pages 860–867, 2005.
- [29] B. Su, S. Lu, and C. Tan. Blurred image region detection and classification. In *Proc. ACM Multimedia*, pages 1397–1400, 2011.
- [30] G. Watson. *Statistics on spheres*, 1983.
- [31] S. Zhong, Y. Liu, and Y. Liu. Bilinear deep learning for image classification. In *Proc. ACM Multimedia*, pages 343–352, 2011.