2013 International Conference on Signal-Image Technology & Internet-Based Systems

High-frequency Restoration Using Deep Belief Nets for Super-resolution

Toru Nakashika
Graduate School of System Informatics
Kobe University
1-1 Rokkodai, Kobe, Japan
nakashika@me.cs.scitec.kobe-u.ac.jp

Tetsuya Takiguchi, Yasuo Ariki
Organization of Advanced Science and Technology
Kobe University
1-1 Rokkodai, Kobe, Japan
takigu@kobe-u.ac.jp, ariki@kobe-u.ac.jp

***Abstract*—Super-resolution technology, which restores high-frequency information given a low-resolved image, has attracted much attention recent years. Various super-resolution algorithms were proposed so far: example-based approach, sparse-coding-based, GMM (Gaussian Mixture Model), BPLP (Back Projection for Lost Pixels), and so on. Most of these statistical approaches rely on the training (or just preparing) of the correspondence relationships between low-resolved/high-resolved images. In this paper, we propose a novel super-resolution method that is based on a statistical model but does not require any pairs of low and high-resolved images in the database. In our approach, Deep Belief Bets are used to restore high-frequency information from a low-resolved image. The idea is that only using high-resolved images, the trained networks seek the high-order dependencies among the observed nodes (each spatial frequency: e.g., high and low frequencies). Experimental results show the high performance of our proposed method.**

***Keywords*-super-resolution; deep-learning; deep-belief-nets; image-restoration;**

I. Introduction

The resolution of the digital camera installed in a cellular phone has dramatically improved in recent years. On the other hand, due to price competition, the need to reduce the cost of the image sensor has become a serious problem, so that the technology to produce high-resolution images using digital image processing is attracting much attention. In general, low-resolution images are enlarged by using interpolation techniques, such as linear or bicubic interpolation. The interpolation methods, however, decrease the resolution of the images because their edge information is lost. For appropriate enlarging, it is necessary to restore the high-frequency components of the image. Such techniques are called "super-resolution", one of the most actively-studied topics on computer vision in recent years. Super-resolution techniques restore the original image from the observed image that has lost its high-frequency components for some reason.

Super-resolution techniques are generally divided into two approaches: example-based methods and linear regression methods. Example-based methods [1], [2] simply use pairs of low-resolution and high-resolution patches for the reconstruction. In this approach, a low-resolved input image is decomposed into patches, each of which is compared with the patches in the database and replaced with the corresponding high-resolved patch. Although this approach produces relatively less-deteriorated images, it is not based on any statistical models and lacks versatility.

Linear regression techniques solve the linear problem of $y = L * x$, where x, y and L are a low-resolution image, a high-resolution image, and a degrading filter, respectively, and $*$ denotes convolution. Generally, it is impossible to obtain the exact high-resolved image because the filter is not known. To approximate this filter, various approaches have been proposed so far: a sparse-coding method [3], [4], [5], [6], total-variation regularization [7], using eigen-space BPLP (Back Projection for Lost Pixels) [8], a Lucy-Richardson method [9], an MRF (Markov Random Field) -based approach [10], [11], a GMM (Gaussian Mixture Model) -based approach [13], a NN (Neural Networks) -based approach [12], and so on. Some of these statistical approaches rely on the training (or just preparing) of the correspondence relationships between low-resolved/high-resolved images. Therefore, if one wants to enlarge an image with the desired scale, the relationships between the low and high resolution with that scale need to be trained beforehand.

Meanwhile, Hinton et el. introduced an effective training algorithm of Deep Belief Nets (DBNs) in 2006 [14], and the use of DBNs rapidly spread in the field of signal processing with great success. DBNs and related models have been, for example, used for hand-written character recognition [14], 3-D object recognition [15], machine transliteration [16], and speech recognition tasks [17]. DBNs are probabilistic generative models that are composed of multiple layers of stochastic latent variables, and have a greedy layer-wise unsupervised learning algorithm. DBNs are not only used for classification tasks, but also for the completion of an image or for collaborative filtering. Eslami et el. adopted a type of DBNs (called SapeBM) to complete the missing region in an image [18]. Salakhutdinov et el. used 2-layer DBNs (i.e., Restricted Boltzmann Machines; RBMs) for collaborative filtering [19], which has the benefit of the DBNs dealing with missing data.

In this paper, we propose a novel super-resolution method using DBNs to restore the missing high-frequencies, motivated by the above-mentioned characteristics of DBNs. In our approach, a low-resolved image is first scaled up to the prescribed size by using bicubic interpolation, and the high-frequency information is estimated by inference of trained DBNs. The networks are trained only using high-resolved image patches in a

978-1-4799-3211-5/13 $31.00 © 2013 IEEE
DOI 10.1109/SITIS.2013.18

multiple-layer-wise unsupervised manner, so as to find the deep relational connections between spatial frequencies. Thus, we expect that the self-trained DBNs capture the high-order dependencies of low-frequencies and high-frequencies, and complete the high-frequency components of a low-resolved image, assuming that the low-frequency components are the same.

II. Overview

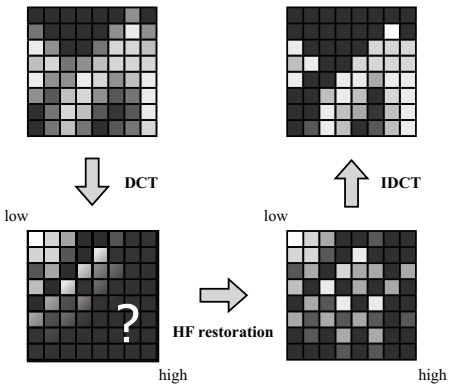


Figure 1. Process of high-frequency restoration.

Generally, an interpolated low-resolved image lacks its spatial high-frequency components. In other words, if the high-frequency components are restored while allowing the low-frequency components to remain, the image is high-resolved. Therefore, it can be regarded as a completion problem of missing data (high-frequency components). This conceptual overview is described in Fig. 1, showing that the high-frequency components of each image patch are restored in the spatial frequency domain transformed by DCT.

We also present a system overview of our proposed method of super-resolution using Deep Belief Nets (DBNs) in Fig. 2. The system has 2 phases of training and restoring. In the training stage, each high-resolved (HR) training image is divided into patches with the size of $P \times P$. Using 2-dimensional DCT, each patch is transformed into the spatial frequency domain. These 2D-DCT coefficients are used for the training of DBNs. In the restoring phase, a low-resolved (LR) image is enlarged by bicubic interpolation and divided into patches with the same size as the training data ($P \times P$). Each patch is transformed to the frequency domain just like the training phase, and fed to the trained DBNs to infer the missing high-frequency components. We give a detailed explanation of this restoring process in subsection III-B. Finally, each output is brought back to the spatial domain using the inverse of 2D-DCT in order to obtain the high-resolved image.

III. Deep Belief Nets

In this paper, we employ Deep Belief Nets (DBNs) for capturing the co-occurrence relationships among DCT

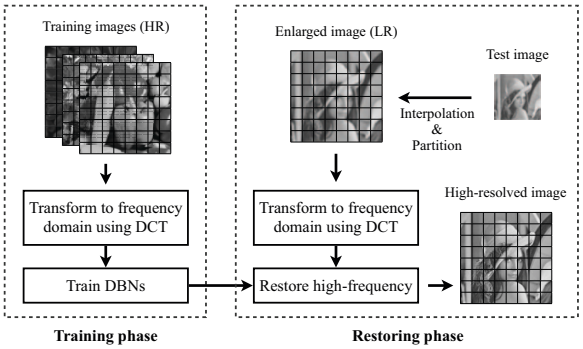


Figure 2. System flowchart of our proposed method.

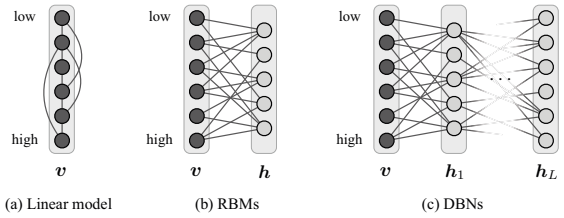


Figure 3. The ways of connections among frequencies with 3 different models.

coefficients based on joint probability. Once the networks are constructed, the lost high-frequency components can be restored based on the co-occurrence.

First, we will briefly give an explanation why we use DBNs. High-frequency restoration based on linear regression models, including sparse-coding, MRF, and BPLP, has the direct connections (potentials or weights) between low and high frequencies $\boldsymbol{v}$ (Fig. 3 (a)). On the other hand, Restricted Boltzmann Machines (RBMs) has no direct connections within the frequencies (visible units $\boldsymbol{v}$), but pairwise connections with independent hidden units $\boldsymbol{h}$ (Fig. 3 (b)). These constraints bring non-linear higher-order dependencies between the visible units. DBNs stack multiple layers of RBMs to get deep architecture (Fig. 3 (c)). Therefore, it is expected that DBNs can capture even higher-order connections between the frequencies. Technically, each stack of RBMs in the DBNs is not a bidirectional model except for the highest layer; however, the architecture is approximately regarded as being a bidirectional model in this paper.

A. Training the Networks

Let us begin with a review of the training method of RBMs before talking about DBNs. In the literature of RBMs, the joint probability $p(\boldsymbol{v}, \boldsymbol{h})$ of real-valued visible units $\boldsymbol{v} = [v_1, \cdots, v_I]^T, v_i \in \mathcal{N}(0,1)$ (note that $I = P^2$, and the training data should be first normalized for each dimension to have zero mean and unit variance) and binary-valued hidden units $\boldsymbol{h} = [h_1, \cdots, h_J]^T, h_j \in \{0,1\}$ is

defined as:

$$p(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{Z}\exp(-E(\boldsymbol{v},\boldsymbol{h})) \quad (1)$$

$$E(\boldsymbol{v},\boldsymbol{h}) = \frac{1}{2}|\boldsymbol{v}|^2 - \boldsymbol{c}^T\boldsymbol{h} - \boldsymbol{v}^T\boldsymbol{W}\boldsymbol{h} \quad (2)$$

$$Z = \sum_{\boldsymbol{v},\boldsymbol{h}}\exp(-E(\boldsymbol{v},\boldsymbol{h})) \quad (3)$$

where, $\boldsymbol{W} \in \mathbb{R}^{I\times J}$ and $\boldsymbol{c} \in \mathbb{R}^{J\times 1}$ are a weight-parameter matrix between visible units and hidden units, and a bias vector of hidden units, respectively.

Since there are no connections between visible units or between hidden units, the conditional probabilities $p(\boldsymbol{h}|\boldsymbol{v})$ and $p(\boldsymbol{v}|\boldsymbol{h})$ form simple equations as follows:

$$p(h_j = 1|\boldsymbol{v}) = \sigma(c_j + \boldsymbol{v}^T\boldsymbol{W}_{:j}) \quad (4)$$

$$p(v_i|\boldsymbol{h}) = \mathcal{N}(\boldsymbol{W}_{i:}\boldsymbol{h}, 1) \quad (5)$$

where $\boldsymbol{W}_{:j}$ and $\boldsymbol{W}_{i:}$ denote the j-th column vector and the i-th row vector, respectively. $\sigma(x)$ indicates sigmoid function, i.e. $\sigma(x) = 1/(1+\exp(-x))$.

For the parameter estimation, the log likelihood of visible units is used as an evaluation function. Differentiating partially with respect to each parameter, we obtain:

$$\frac{\partial \log p(\boldsymbol{v})}{\partial w_{ij}} = \langle v_i h_j\rangle_{data} - \langle v_i h_j\rangle_{model} \quad (6)$$

$$\frac{\partial \log p(\boldsymbol{v})}{\partial c_j} = \langle h_j\rangle_{data} - \langle h_j\rangle_{model} \quad (7)$$

where, $\langle\cdot\rangle_{data}$ and $\langle\cdot\rangle_{model}$ indicate expectations of input data and the inner model, respectively. However, it is generally hard to compute the second term. Typically, expectation of the reconstructed data computed by Eqs. (4) and (5) is alternatively used [14]. Using Eqs. (6) and (7), each parameter can be updated by stochastic gradient descent.

In the training of DBNs, the hidden units of the current stack are regarded as visible units in the next layer. In other words, the hidden units computed as a conditional probability $p(\boldsymbol{h}|\boldsymbol{v})$ in Eq. (4) are fed to the following RBMs, and trained in the similar way. This time, the conditional probability $p(\boldsymbol{v}|\boldsymbol{h})$ is computed by

$$p(v_i = 1|\boldsymbol{h}) = \sigma(b_i + \boldsymbol{W}_{i:}\boldsymbol{h}) \quad (8)$$

introducing a bias vector of visible units $\boldsymbol{b} \in \mathbb{R}^{I\times 1}$, because the visible units propagated from the previous RBMs have binary values. This procedure is repeated layer-by-layer until the highest layer.

B. Inferring the High-frequencies

The basic idea of the restoration of high-frequency components using DBNs is depicted in Fig. 4. Given a low-resolved bicubic-interpolated image, the DCT low-frequency coefficients are first emphasized to raise the damped parts caused by the interpolation, and fed into the trained DBNs. Note that the high-frequency coefficients are almost zero at this point. Starting with the upper left of Fig. 4, the input coefficients $\boldsymbol{u}^0 = [u_1^0, \cdots, u_I^0]^T$ are propagated to the following layers in order by Eq. (4),

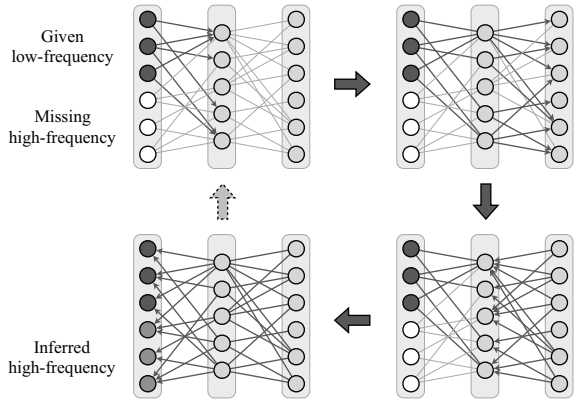


Figure 4. The process of high-frequency restoration using DBNs (2 hidden layers in this example).

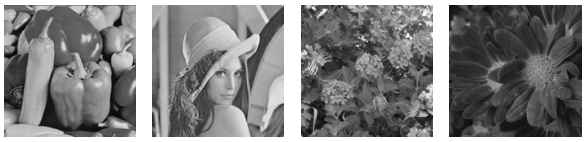


Figure 5. Four input images. Only (I) is used for training, and the others for testing.

and back-propagated by Eqs. (5) and (8). Then, we obtain the predicted values including high-frequency components $\boldsymbol{r}^0 = [r_1^1, \cdots, r_I^1]^T$ (lower left in Fig. 4). In our repetitive approach, we input the obtained output vector to the DBNs again and repeat the same procedure Q times. The output vector $\boldsymbol{u}^q = [u_1^q, \cdots, u_I^q]^T$ at q-th iteration ($q = 1, \cdots, Q$) is, in this work, given by

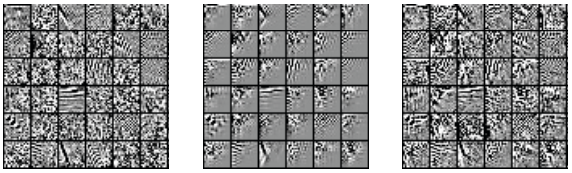
$$u_i^q = \begin{cases} \beta \cdot r_i^{q-1} & (i \geq I/s) \\ u_i^0 & (i < I/s) \end{cases} \quad (9)$$

where s and β denote the scaling size of the interpolation and a high-frequency emphasis parameter, respectively, and i ($i = 1, \cdots, I$) is the index of 2D-DCT coefficients in zig-zag scan. Eq. (9) means that high-frequency components show up gradually without changing the low-frequency components, considering that low-frequency components are the same even when high-resolved.

IV. Experiments

A. Setup

For the training of Deep Belief Nets (DBNs), we used image (I) shown in Fig. 5, whose size is 512×512. We partitioned the image to have the size on a side of $P = 16$ with quarterly-overlapping. Each patch (in total 15625 patches) was transformed by 2-dimensional DCT, normalized, and then fed to DBNs. We trained DBNs with a learning rate of 0.01 for 500 epochs, which have 2-hidden layers, 400 hidden units for the first layer and 200 hidden units for the second layer. In order to examine the trained DBNs, we show in Fig. 6 some examples of



(a) Training examples (b) Bicubic (c) Proposed

Figure 6. Examples of normalized 2D-DCT coefficients.

the training data (a), bicubic-interpolated DCT coefficients (b), and restored data from (b) using our proposed method (c). As shown in Fig. 6 (b), while each data lacks its high-frequency coefficients. In our approach, the lost high-frequency components were restored. High-frequencies of some patches were estimated incorrectly, but most of the patches capture the overall patterns of the training patches (for example, straight lines or dotted patterns).

For testing, 3 images (Fig. 5(II)(III)(IV)) were reduced by half ($s = 2$) from 512×512 in the horizontal and vertical directions, and enlarged by two times using our proposed method. We used $Q = 2$ and $\beta = 1.5$ as the parameters for the restoration, which performed best.

To evaluate the efficacy of our method, we compared it with 2 conventional methods (sparse-coding [3] and GMM [13]) and bicubic interpolation with 2 measures (PSNR and SSIM [20]). Given an original image $\mathbf{Y}$ (high-resolved image) and its processed image $\mathbf{E}$, PSNR and SSIM measure the quality of the processed image. The larger the values of PSNR and SSIM are, the higher the quality of the images is supposed to be. PSNR and SSIM are defined as follows:

$$PSNR = 10 \log_{10} \frac{255^2}{|\mathbf{E} - \mathbf{Y}|_2^2} \quad (10)$$

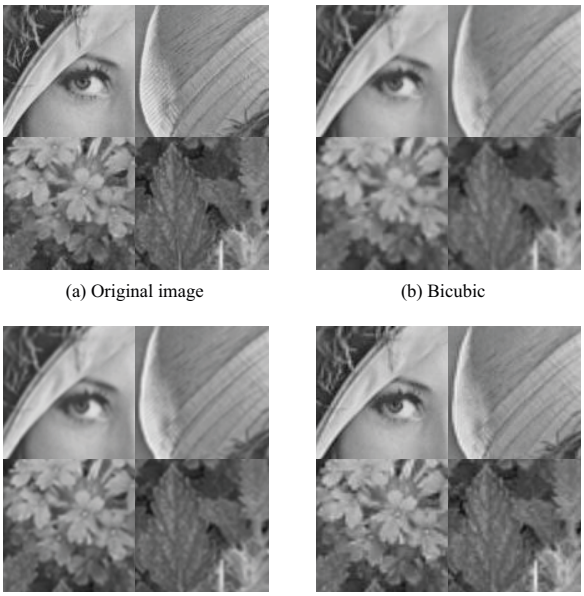
$$SSIM = \frac{(2\mu_E\mu_Y)(2\sigma_{EY})}{(\mu_E^2 + \mu_Y^2)(\sigma_E^2 + \sigma_Y^2)} \quad (11)$$

where μ_Y and μ_E are the averages over the images $\mathbf{Y}$ and $\mathbf{E}$, respectively, σ_Y and σ_E are the variances of $\mathbf{Y}$ and $\mathbf{E}$, respectively, and σ_{EY} is the covariance of $\mathbf{Y}$ and $\mathbf{E}$.

For reference, we also compared within our methods to different architecture of DBNs: 1-layer, 400 hidden units (i.e. RBMs).

B. Results and Discussion

Table I summarizes the experimental results, and Fig. 7 compares super-resolved images by bicubic interpolation, GMM, and our proposed method. As shown in Table I, the proposed method using DBNs performed best for each test image with either measure. Furthermore, 2-hidden-layer DBNs (Proposed(DBNs)) outperformed 1-hidden-layer DBNs (Proposed(RBMs)). In Fig. 7, we can also see that the edges are better emphasized by our method than bicubic and GMM. The architecture of the deep DBNs captures higher-order dependencies between low and high frequencies better than the other methods including shallow DBNs, and we consider that this ends up with the preferable results.



(a) Original image (b) Bicubic

(c) GMM (d) Proposed

Figure 7. Super-resolved examples of the image (II) and (III) (in the first and second rows, respectively) by various methods.

Table I
COMPARISON OF SUPER-RESOLUTION METHODS USING PSNR AND SSIM.

Image	Method	PSNR	SSIM
(II)	Bicubic	36.43	0.8816
	Sparse-Coding	37.71	0.9084
	GMM	37.98	0.9272
	Proposed(RBMs)	38.52	0.9289
	Proposed(DBNs)	**38.60**	**0.9308**
(III)	Bicubic	33.07	0.8015
	Sparse-Coding	34.20	0.8608
	GMM	35.59	0.9154
	Proposed(RBMs)	35.73	0.8639
	Proposed(DBNs)	**37.60**	**0.9067**
(IV)	Bicubic	38.15	0.8977
	Sparse-Coding	39.68	0.9240
	GMM	40.83	0.9460
	Proposed(RBMs)	40.40	0.9452
	Proposed(DBNs)	**41.31**	**0.9548**

V. CONCLUSION

In this work, we proposed the use of Deep Belief Nets (DBNs) to tackle super-resolution, replacing the task with the completion problem of the missing data. In our approach, the missing high-frequency components in a low-resolved image are restored using self-trained DBNs in the spatial frequency domain. In our experiments, we showed the efficacy of the proposed method, in comparison to conventional methods. Future work will include the use of Deep Boltzmann Machines instead of DBNs to improve the reconstruction accuracy, which have a deep bidirectional model.

REFERENCES

[1] W. T. Freeman, T. R. Jones, and E. C. Pasztor, "Example-Based Super-Resolution," IEEE Computer Graphics and Applications, vol. 22, no. 2, pp. 56–65, 2002.

[2] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning Low-Level Vision," International Journal of Computer Vision, vol. 40, no. 1, pp. 25–47, 2000.

[3] A. T. Mario Figueiredo Michael Elad, and M. Yi , "On the role of sparse and redundant representations in image processing," Proceedings of the IEEE, vol. 98, no. 6, pp. 972–982, 2010.

[4] Z. Lin, "Bilevel sparse coding for coupled feature spaces," IEEE Conference on Computer Vision and Pattern Recognition, pp. 2360–2367, 2012.

[5] H. Li, Q. Hairong, Z. Russell, "Beta process joint dictionary learning for coupled feature spaces with application to single image super-resolution," Computer Vision and Pattern Recognition, pp. 23–28, 2013.

[6] J. Gao, Y. Guo, and M. Yin, "Restricted boltzmann machine approach to couple dictionary training for image super-resolution," IEEE International Conference on Image Processing, pp. 499–503, 2013.

[7] S. D. Babacan, "Total variation super resolution using a variational approach," 15th IEEE International Conference on Image Processing, pp. 641–644, 2008.

[8] H. Kawano, N. Suetake, and H. Maeda, "Image Enlargement with Lost High-Frequency Components Estimation Using Clustered Eigenspace-BPLP," Image and Video Technology (PSIVT), pp. 404–407, 2010.

[9] S. B. Kasturiwala and S. A. Ladhake, "Superresolution: A Novel Application to Image Restoration," International Journal of Computational Science and Engineering, vol. 5, no. 5, pp. 1659–1664, 2010.

[10] R. Deepu and C. Subhasis, "An MRF-Based Approach to Generation of Super-Resolution Images from Blurred Observations," Journal of Mathematical Imaging and Vision, vol. 16, pp. 5–15, 2002.

[11] Z. Liangpei, Z. Hongyan, S. Huanfeng, and L. Pingxiang, "A super-resolution reconstruction algorithm for surveillance images," Journal of Signal Processing, vol. 90, pp. 848–859, 2010.

[12] Y. Huang and Y. Long, "Super-resolution using neural networks based on the optimal recovery theory," Journal of Computational Electronics, vol. 5, issue 4, pp. 275–281, 2006.

[13] Y. Ogawa, Y. Ariki, and T. Takiguchi, "Super-resolution by GMM based conversion using self-reduction image," Proceedings of International Conference on Acoustics, Speech and Signal Processing, pp. 1285–1288, 2012.

[14] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," Neural Computation, vol. 18, pp. 1527–1554, 2006.

[15] V. Nair and G. Hinton, "3-D object recognition with deep belief nets," Advances in Neural Information Processing Systems 22, pp. 1339–1347, 2009.

[16] T. Deselaers, S. Hasan, O. Bender, and H. Ney, "A deep learning approach to machine transliteration," in Proc. EACL Workshop on Statistical Machine Translation, 2009, pp. 233–241.

[17] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic Modeling using Deep Belief Networks," IEEE Trans. on Audio, Speech, and Language Procesing, vol. 20, no. 1, pp. 14–22, 2012.

[18] S. M. Eslami, N. Heess, and J. Winn, "The Shape Boltzmann Machine: a Strong Model of Object Shape," IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[19] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in Proc. International Conference on Machine Learning, pp. 791–798, 2007.

[20] Z. Wang, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Trans. on Image Processing, vol. 13, no. 4, pp. 600–612, 2004.