Deep Markov Random Field for Image Modeling

Zhirong Wu(✉), Dahua Lin, and Xiaoou Tang

The Chinese University of Hong Kong, Sha Tin, Hong Kong
{zhirong,dhlin,xtang}@ie.cuhk.edu.hk

Abstract. Markov Random Fields (MRFs), a formulation widely used in generative image modeling, have long been plagued by the lack of expressive power. This issue is primarily due to the fact that conventional MRFs formulations tend to use *simplistic* factors to capture local patterns. In this paper, we move beyond such limitations, and propose a novel MRF model that uses fully-connected neurons to express the complex interactions among pixels. Through theoretical analysis, we reveal an inherent connection between this model and recurrent neural networks, and thereon derive an approximated feed-forward network that couples multiple RNNs along opposite directions. This formulation combines the expressive power of deep neural networks and the cyclic dependency structure of MRF in a unified model, bringing the modeling capability to a new level. The feed-forward approximation also allows it to be efficiently learned from data. Experimental results on a variety of low-level vision tasks show notable improvement over state-of-the-arts.

Keywords: Generative image model · MRF · RNN

1 Introduction

Generative image models play a crucial role in a variety of image processing and computer vision tasks, such as denoising [1], super-resolution [2], inpainting [3], and image-based rendering [4]. As repeatedly shown by previous work [5], the success of image modeling, to a large extent, hinges on whether the model can successfully capture the spatial relations among pixels.

Existing image models can be roughly categorized as *global models* and *low-level models*. Global models [6–8] usually rely on compressed representations to capture the global structures. Such models are typically used for describing objects with regular structures, *e.g.* faces. For generic images, low-level models are more popular. Thanks to their focus on local patterns instead of global appearance, low-level models tend to generalize much better, especially when there can be vast variations in the image content.

Over the past decades, *Markov Random Fields (MRFs)* have evolved into one of the most popular models for low-level vision. Specifically, the clique-based

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46484-8_18) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part VIII, LNCS 9912, pp. 295–312, 2016.
DOI: 10.1007/978-3-319-46484-8_18

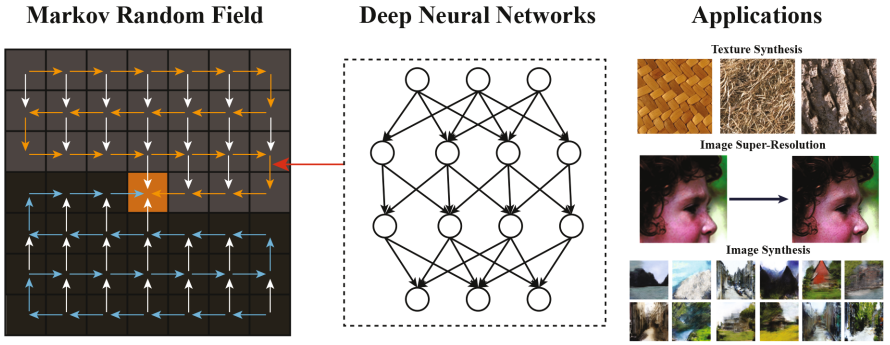

Fig. 1. We present a new class of markov random field models whose potential functions are expressed by powerful deep neural networks. We show applications of the model on texture synthesis, image super-resolution and image synthesis.

structure makes them particularly well suited for capturing local relations among pixels. Whereas MRFs as a generic mathematical framework are very flexible and provide immense expressive power, the performance of many MRF-based methods still leaves a lot to be desired when faced with challenging conditions. This occurs due to the widespread use of *simplistic* potential functions that largely restrict the expressive power of MRFs (Fig. 1).

In recent years, the rise of *Deep Neural Networks (DNN)* has profoundly reshaped the landscape of many areas in computer vision. The success of DNNs is primarily attributed to its unparalleled expressive power, particularly their strong capability of modeling complex variations. However, DNNs in computer vision are mostly formulated as end-to-end convolutional networks (CNN) for classification or regression. The modeling of local interactions among pixels, which is crucial for many low-level vision tasks, has not been sufficiently explored.

The respective strengths of MRFs and DNNs inspire us to explore a new approach to low-level image modeling, that is, to bring the expressive power of DNNs to an MRF formulation. Specifically, we propose a generative image model comprised of a grid of *hidden states*, each corresponding to a pixel. These latent states are connected to their neighbors – together they form an MRF. Unlike in classical MRF formulations, we use fully connected layers to express the relationship among these variables, thus substantially improving the model's ability to capture complex patterns.

Through theoretical analysis, we reveal an inherent connection between our MRF formulation and the RNN [9], which opens an alternative way to MRF formulation. However, they still differ fundamentally: the dependency structure of an RNN is *acyclic*, while that of an MRF is *cyclic*. Consequently, the hidden states cannot be inferred in a single *feed-forward* manner as in a RNN. This posts a significant challenge – how can one derive the back-propagation procedure without a well-defined forward function?

Our strategy to tackle this difficulty is to *unroll an iterative inference procedure into a feed-forward function*. This is motivated by the observation that while the inference is iterative, each cycle of updates is still a feed-forward procedure. Following a carefully devised scheduling policy, which we call the *Coupled Acyclic Passes (CAP)*, the inference can be unrolled into multiple RNNs operating along opposite directions that are coupled together. In this way, local information can be effectively propagated over the entire network, where each hidden state can have a complete picture of its context from all directions.

The primary contribution of this work is a new generative model that unifies MRFs and DNNs in a novel way, as well as a new learning strategy that makes it possible to learn such a model using mainstream deep learning frameworks. It is worth noting that the proposed method is generic and can be adapted to a various problems. In this work, we test it on a variety of low-level vision tasks, including texture synthesis, image super-resolution, and image synthesis.

2 Related Works

In this paper, we develop a generative image model that incorporates the expressive power of deep neural networks with an MRF. This work is related to several streams of research efforts, but moves beyond their respective limitations.

Generative image models. Generative image models generally fall into two categories: parametric models and non-parametric models. *Parametric models* typically use a compressed representation to capture an image's global appearance. In recent years, deep networks such as autoencoders [10] and adversarial networks [11,12] have achieved substantial improvement in generating images with regular structures such as faces or digits. *Non-parametric models*, including *pixel-based sampling* [13–15] and *patch-based sampling* [16–18], instead rely on a large set of exemplars to capture local patterns. Whereas these methods can produce high quality images with local patterns directly sampled from realistic images. Exhaustive search over a large exemplar set limits their scalability and often leads to computational difficulties. Our work draws inspiration from both lines of work. By using DNNs to express local interactions in an MRF, our model can capture highly complex patterns while maintaining strong scalability.

Markov random fields. For decades, MRFs have been widely used for low-level vision tasks, including texture synthesis [19], segmentation [20,21], denoising [1], and super-resolution [2]. Classical MRF models in earlier work [22] use simple hand-crafted potentials (*e.g.*, Ising models [23], Gaussian MRFs [24]) to link neighboring pixels. Later, more flexible models such as FRAME [25] and Fields of Experts [26] were proposed, which allow the potential functions to be learned from data. However, in these methods, the potential functions are usually parameterized as a set of linear filters, and therefore their expressive power remains very limited.

Recurrent neural networks. *Recurrent neural networks (RNNs)*, a special family of deep models, use a chain of nonlinear units to capture sequential relations. In computer vision, RNNs are primarily used to model sequential changes

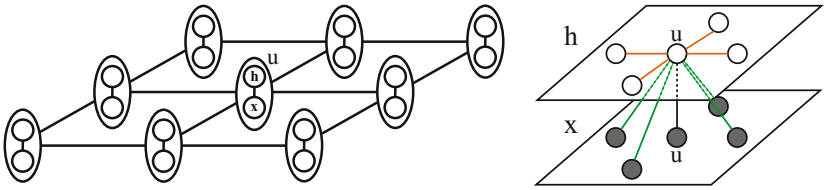

Fig. 2. Graphical model of deep MRFs. **Left**: The hidden states and the pixels together form an MRF. **Right**: Each hidden state connects to the neighboring states, the neighboring pixels, and the pixel at the same location.

in videos [27], visual attention [28,29], and hand-written digit recognition [30]. Previous work explores multi-dimensional RNNs [31] for scene labeling [32] as well as object detections [33]. The most related work is perhaps the use of 2D RNNs for generating gray-scale textures [34] or color images [35]. A key distinction of these models from ours is that 2D RNNs rely on an *acyclic graphs* to model spatial dependency, *e.g.* each pixel depends only on its left and upper neighbors – this severely limits the spatial coherence. Our model, instead, allows dependencies from all directions via iterative inference unrolling.

MRF and neural networks. Connections between both models have been discussed long ago [36]. With the rise of deep learning, recent work on image segmentation [37,38] uses mean field method to approximate a conditional random field (CRF) with CNN layers. A hybrid model of CNN and MRF has also been proposed for human pose estimation [39]. These works primarily target prediction problems (*e.g.* segmentation) and are not as effective at capturing complex pixel patterns in a purely generative way.

3 Deep Markov Random Field

The primary goal of this work is to develop a generative model for images that can express complex local relationships among pixels while being tractable for inference and learning. Formally, we consider an image, denoted by $\mathbf{x}$, as an *undirected graph* with a grid structure, as shown in Fig. 2 left. Each node u corresponds to a pixel x_u. To capture the interactions among pixels, we introduce, h_u, a hidden variable for each pixel denoting the hidden state corresponding to the pixel x_u. In the graph, each node u has a neighborhood, denoted by $\mathcal{N}_u$. Particularly, we use the *4-connected neighborhood* of a 2D grid in this work.

Joint Distribution. We consider three kinds of dependencies: (1) the dependency between a pixel x_u and its corresponding hidden state h_u, (2) the dependency between a hidden state h_u and a neighbor h_v with $v \in \mathcal{N}_u$, and (3) the dependency between a hidden state h_u and a neighboring pixel x_v. They are respectively captured by factors $\zeta(x_u, h_u)$, $\phi(h_u, h_v)$, and $\psi(h_u, x_v)$. In addition, we introduce a regularization factor $\lambda(h_u)$ for each hidden state, which gives us

the leeway to encourage certain distribution over the state values. Bringing these factors together, we formulate an MRF to express the joint distribution:

$$p(\mathbf{x}, \mathbf{h}) = \frac{1}{Z} \prod_{u \in V} \zeta(x_u, h_u) \prod_{(u,v) \in E} (\phi(h_u, h_v)\psi(h_u, x_v)\psi(h_v, x_u)) \prod_{u \in V} \lambda(h_u). \quad (1)$$

Here, V and E are respectively the set of vertices and that of the edges in the image graph, Z is a normalizing constant. Figure 2 shows it structure.

Choices of Factors. Whereas the MRF provides a principled way to express the dependency structure, the expressive power of the model still largely depends on the specific forms of the factors that we choose. For example, the modeling capacity of classical MRF models are limited by their simplistic factors.

Below, we discuss the factors that we choose for the proposed model. First, the factor $\zeta(x_u, h_u)$ determines how the pixel values are generated from the hidden states. Considering the stochastic nature of natural images, we formalize this generative process as a *Gaussian mixture model (GMM)*. The rationale behind is that pixel values are on a low-dimensional space, where a GMM with a small number of components can usually provide a good approximation to an empirical distribution. Specifically, we fix the number of components to be K, and consider the concatenation of component parameters as the linear transform of the hidden state, $h_u^T \mathbf{Q} = ((\pi_u^c, \mu_u^c, \Sigma_u^c))_{c=1}^K$, where $\mathbf{Q}$ is a weight matrix of model parameters. In this way, the factor $\zeta(x_u, h_u)$ can be written as

$$\zeta(x_u, h_u) \triangleq p_{\text{GMM}}(x_u|h_u) = \sum_{c=1}^{K} \pi_u^c N(x_u|\mu_u^c, \Sigma_u^c). \quad (2)$$

To capture the rich interactions among pixels and their neighbors, we formulate the relational factors $\phi(h_u, h_v)$ and $\psi(h_u, x_v)$ with *fully connected* forms:

$$\phi(h_u, h_v) = \exp\left(h_u^T \mathbf{W} h_v\right), \quad \psi(h_u, x_v) = \exp\left(h_u^T \mathbf{R} x_v\right). \quad (3)$$

Finally, to control the value distribution of the hidden states, we further incorporate a regularization term over h_u, as

$$\lambda(h_u) = \exp\left(-\mathbf{1}^T \eta(h_u)\right) = \exp\left(-\eta(h_u^{(1)}) - \cdots - \eta(h_u^{(d)})\right). \quad (4)$$

Here, η is an element-wise nonlinear function and d is the dimension of h_u. In summary, the use of GMM in $\zeta(x_u, h_u)$ effectively accounts for the variations in pixel generation, the fully-connected factors $\phi(h_u, h_v)$ and $\psi(h_u, x_v)$ enable the modeling of complex interactions among neighbors, while the regularization term $\lambda(h_u)$ provides a way to explicitly control the distribution of hidden states. Together, they substantially increase the capacity of the MRF model.

Inference of Hidden States. With this MRF formulation, the posterior distribution of the hidden state h_u, conditioned on all other variables, is given by

$$p\left(h_u \mid x_u, x_{\mathcal{N}_u}, h_{\mathcal{N}_u}\right) \propto \zeta(x_u, h_u)\lambda(h_u) \cdot \prod_{v \in \mathcal{N}_u} \phi(h_u, h_v)\psi(h_u, x_v). \quad (5)$$

Here, h_u depends on its neighboring states, the corresponding pixel values, as well as that of its neighbors. Since the pixel x_u and its neighboring pixels $x_{\mathcal{N}_u}$ are highly correlated, to simplify our later computations, we approximate the posterior distribution as,

$$p\left(h_u \mid x_u, x_{\mathcal{N}_u}, h_{\mathcal{N}_u}\right) \simeq p\left(h_u \mid x_{\mathcal{N}_u}, h_{\mathcal{N}_u}\right) \propto \lambda(h) \prod_{v \in \mathcal{N}_u} \phi(h, h_v)\psi(h, x_v). \quad (6)$$

We performed numerical simulations for this approximation. They are indeed very close to each other, as illustrated in Fig. 3. Consequently, the MAP estimate of h_u can be *approximately* computed from its neighbors. It turns out that this optimization problem has an analytic solution given by,

$$\tilde{h}_u = \sigma\left(\sum_{v \in \mathcal{N}_u} \mathbf{W} h_v + \mathbf{R} x_v\right). \quad (7)$$

Here, σ is an element-wise function that is related to η as $\sigma^{-1}(z) = \eta'(z)$, where η' is the first-order derivative *w.r.t.* η, and σ^{-1} the inverse function of σ.

Connections to RNNs. We observe that Eq. (7) has a form that is similar to the feed-forward computations in *Recurrent Neural Networks (RNN)* [9]. In this sense, we can view the feed-forward RNN as an MAP inference process for MRF models. Particularly, given the RNN computations in the form of Eq. (7), one can formulate an MRF as in Eq. (1), where regularization function η can be derived from σ according to the relation $\sigma^{-1}(z) = \eta'(z)$, as

$$\eta(h) = \int_b^h \sigma^{-1}(z)dz + C. \quad (8)$$

Here, b is the minimum of the domain of h, which can be $-\infty$, and C is an arbitrary constant. This connection provides an alternative way to formulate an MRF model. More importantly, in this way, RNN models that have been proven to be successful can be readily transferred to an MRF formulation. Figure 3 shows the regularization functions $\eta(h)$ corresponding to popular activation functions in RNNs, such as *sigmoid* and *ReLU* [40].

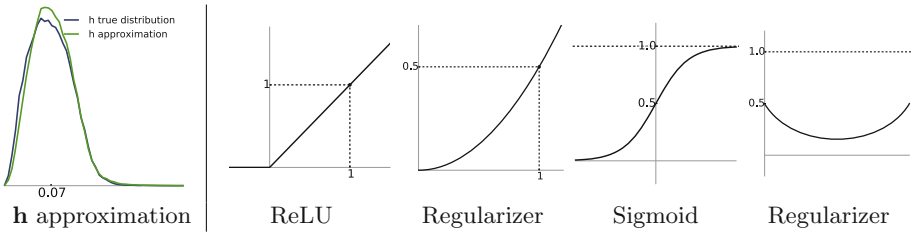

Fig. 3. Left shows the numerical simulation of approximated inference for the hidden variables. **Right** shows the ReLU, sigmoid activation function and their corresponding regularizations for the hidden variables.

4 Learning via Coupled Recurrent Networks

Except for special cases [41], inference and learning on MRFs is generally intractable. Conventional estimation methods [8,42,43] either take overly long time to train or tend to yield poor estimates, especially for models with a high-dimensional parameter space. In this work, we consider an alternative approach to MRF learning, which allows us to draw on deep learning techniques [44,45] that have been proven to be highly effective [40].

Variational Learning Principle. Estimation of probabilistic models based on the *maximum likelihood* principle is often intractable when the model contains hidden variables. *Expectation-maximization* [46] is one of the most widely used ways to tackle this problem, which iteratively calculates the posterior distribution of $\mathbf{h}_i$ (in E-steps) and then optimizes $\boldsymbol{\theta}$ (in M-steps) as

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ \frac{1}{n} \sum_{i=1}^{n} \mathrm{E}_{p(\mathbf{h}_i|\mathbf{x}_i,\boldsymbol{\theta})} \left\{ \log p(\mathbf{x}_i, \mathbf{h}_i|\boldsymbol{\theta}) \right\}. \tag{9}$$

Here, $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{Q}, \mathbf{R}\}$ is the model parameter, $\mathbf{x}_i$ is the i-th image, and $\mathbf{h}_i$ is the corresponding hidden state. As exact computation of this posterior expectation is intractable, we approximate it based on $\tilde{\mathbf{h}}_i$, the MAP estimate of $\mathbf{h}_i$, as below:

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \ \frac{1}{n} \sum_{i=1}^{n} \log p(\mathbf{x}_i|\tilde{\mathbf{h}}_i, \boldsymbol{\theta}), \ \text{with } \tilde{\mathbf{h}}_i \triangleq f(\mathbf{x}_i, \boldsymbol{\theta}). \tag{10}$$

This is the *learning objective* of our model. Here, f is the function that *approximately* infers the latent state $\tilde{\mathbf{h}}_i$ given an observed image $\mathbf{x}_i$. When the posterior distribution $p(\mathbf{h}_i|\mathbf{x}_i, \boldsymbol{\theta})$ is highly concentrated, which is often the case in vision tasks, this is a good approximation. For an image $\mathbf{x}$, $\log p(\mathbf{x}|\tilde{\mathbf{h}}, \boldsymbol{\theta})$ can be further expanded as a sum of terms defined on individual pixels:

$$\log p(\mathbf{x}|\tilde{\mathbf{h}}, \boldsymbol{\theta}) = \sum_u \log p_{\mathrm{GMM}}(x_u|\tilde{\mathbf{h}}) = \sum_u \log \sum_{c=1}^{K} \pi_u^c N(x_u|\tilde{\mu_u^c}, \Sigma_u^c), \tag{11}$$

where $\tilde{\mu_u^c} = \mu_u^c + \Sigma_u^c (\sum_v \mathbf{h}_v^T)\mathbf{R}$. For our problem, this learning principle can be interpreted in terms of encoding/decoding – the hidden states $\tilde{\mathbf{h}} = f(\mathbf{x}, \boldsymbol{\theta})$ can be understood as an representation that encodes the observed patterns in an image $\mathbf{x}_i$, while $\log p(\mathbf{x}|\tilde{\mathbf{h}}, \boldsymbol{\theta})$ measures how well $\tilde{\mathbf{h}}$ explains the observations.

Coupled Acyclic Passes. In the proposed model, the dependencies among neighbors are *cyclic*. Hence, the MAP estimate $\tilde{\mathbf{h}} = f(\mathbf{x}, \boldsymbol{\theta})$ cannot be computed in a single forward pass. Instead, Eq. (7) needs to be applied across the graph in multiple iterations. Our strategy is to unroll this iterative inference procedure into multiple feed-forward passes along opposite directions, such that these passes together provide a complete context to each local estimate.

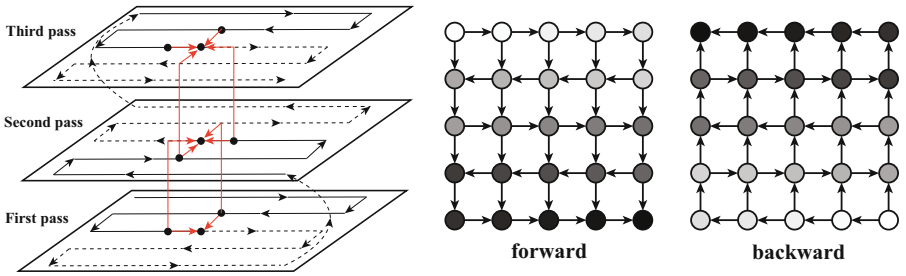

Fig. 4. Coupled acyclic passes. We decouple an undirected cyclic graph into two directed acyclic graphs with each one allowing feed-forward computation. Inference is performed by alternately traversing the two acyclic graphs, while coupling their information at each step.

Specifically, we decompose the underlying dependency graph $G = (V, E)$, which is undirected, into two *acyclic directed graphs* $G^f = (V, E^f)$ and $G^b = (V, E^b)$, as illustrated in Fig. 4, such that each undirected edge $\{u, v\} \in E$ corresponds uniquely to an edge $(u, v) \in E^f$ and an opposite edge $(v, u) \in E^b$. It can be proved that such a decomposition always exists and that for each node $u \in V$, the neighborhood $\mathcal{N}_u$ can be expressed as $\mathcal{N}_u = \mathcal{N}^f(u) \cup \mathcal{N}^b(u)$, where $\mathcal{N}^f(u)$ and $\mathcal{N}^b(u)$ are the set of parents of u respectively along G^f and G^b.

Given such a decomposition, we can derive an iterative computational procedure, where each cycle couples a *forward pass* that applies Eq. (7) along G^f and a *backward pass*[1] along G^b. After the t-th cycle, the state h_u is updated to

$$h_u^{(t)} = \sigma\left(\sum_{v \in \mathcal{N}^f(u)} \left(\mathbf{W} h_v^{(t-1)} + \mathbf{R} x_v \right) + \sum_{v \in \mathcal{N}^b(u)} \left(\mathbf{W} h_v^{(t)} + \mathbf{R} x_v \right) \right). \tag{12}$$

As states above, we have $\mathcal{N}_u = \mathcal{N}^f(u) \cup \mathcal{N}^b(u)$. Therefore, over a cycle, the updated state h_u would incorporate information from all its neighbors. Note that a given graph G can be decomposed in many different ways. In this work, we specifically choose the one that forms the *zigzag* path. The advantage over a simple raster line order is that *zigzag* path traverses all the nodes continuously, so that it conserves spatial coherence by making dependence of each node to all the previous nodes that have been visited before. The forward and backward passes resulted from such decomposition are shown in Fig. 4.

This algorithm has two important properties: First, the acyclic decomposition allows feed-forward computation as in Eq. (7) to be applied. As a result, the entire inference procedure can be viewed as a feed-forward network that couples multiple RNNs operating along different directions. Therefore, it can be learned in a way similar to other deep neural networks, using *Stochastic Gradient Descent (SGD)*. Second, the feedback mechanism embodied by the backward pass

[1] The word *forward* and *backward* here means the sequential order in the graph. They are not *feed-forward* and *back-propagation* in the context of deep neural networks.

facilitates the propagation of local information and thus the learning of long-range dependencies.

Discussions with 2D-RNN. Previous work has explored two-dimensional extensions of RNN [31], often referred to as *2D-RNN*. Such extensions, however, are formulated upon an acyclic graph, and can be considered as a trimmed down version of our algorithm. A major drawback of 2D-RNN is that it scans the image in a raster line order and it is not able to provide a feedback path. Therefore, the inference of each hidden state can only take into account 1/4 of the context, and there is no way to recover from a poor inference. As we will show in our experiments, this may cause undesirable effects. Whereas bidirectional RNNs [47] may partly mitigate this problem, they decouple the hidden states into multiple ones that are independent apriori, which would lead to consistency issues. Recent work [48] also finds it difficult to use in generative modeling.

Implementation Details. For inference and learning, to make the computation feasible, we just take one forward pass and one backward pass. Thus, each node is only updated twice while being able to use the information from all possible contexts. The training patch size varies from 15 to 25 depending on the specific experiment. Overall, if we unroll the full inference procedure, our model[2] is more than thousands of layers deep. We use *rmsprop* [45] for optimization and we don't use dropout for regularization, as we find it oscillates the training.

5 Experiments

In the following experiments, we test the proposed deep MRF on 3 scenarios for modeling natural images. We first study its basic properties on *texture synthesis*, and then we apply it on a prediction problem, *image super-resolution*. Finally, we integrate global CNN models with local deep MRF for *natural image synthesis*.

5.1 Texture Synthesis

The task of texture synthesis is to synthesize new texture images that possess similar patterns and statistical characteristics as a given texture sample. The study of this problem originated from graphics [13,14]. The key to successful texture reproduction, as we learned from previous work, is to effectively capture the local patterns and variations. Therefore, this task is a perfect testbed to assess a model's capability of modeling visual patterns.

Our model works in a purely generative way. Given a sample texture, we train the model on randomly extracted patches of size 25×25, which are larger than most *texels* in natural images. We set $K = 20$, initialize $\mathbf{x}$ and $\mathbf{h}$ to zeros, and train the model with back-propagation along the coupled acyclic graph. With a trained model, we can generate textures by running the RNN to derive the latent states and at the same time sampling the output pixels. As our model is stationary, it can generate texture images of arbitrary sizes.

[2] Code available at https://github.com/zhirongw/deep-mrf.

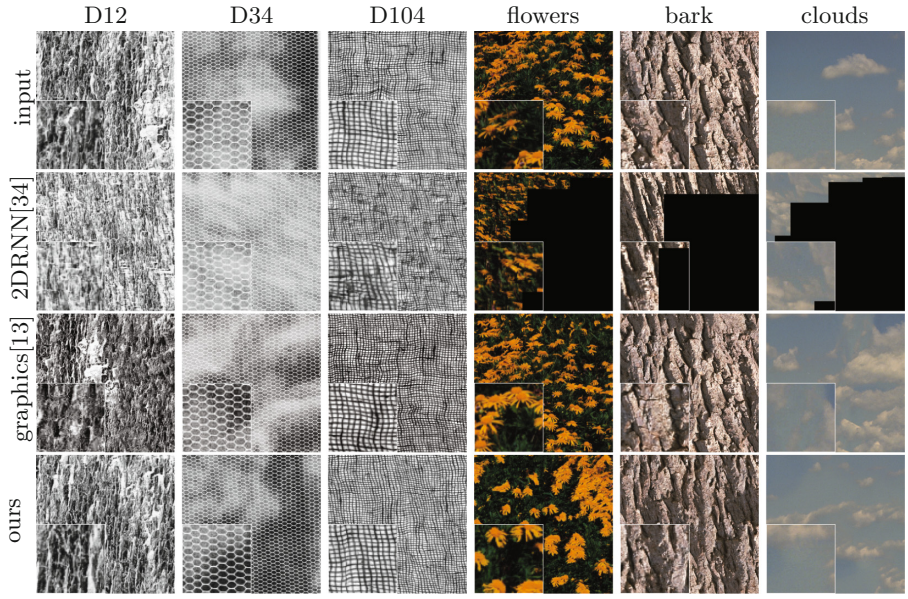

Fig. 5. Texture synthesis results.

We work on two texture datasets, Brodatz [49] for grayscale images, and VisTex [50] for color images. From the results shown in Fig. 5, our synthesis visually resembles to high resolution natural images, and the quality is close to the non-parametric approach [13]. We also compare with the 2D-RNN [34]. As we can see, the results obtained using 2D-RNN, which synthesizes based only on the left and upper regions, exhibit undesirable effects and often evolve into blacks in the bottom-right parts.

Two fundamental parameters control the behaviors of our texture model. The training patch size decides the farthest spatial relationships that could be learned from data. The number of gaussian mixtures control the dynamics of the texture landscape. We analyze our model by changing the two parameters. As shown in Fig. 6, bigger training patch size and bigger number of mixtures consistently improves the results. For non-parametric approaches, bigger patch size would dramatically bring up the computation cost. While for our model, the inference time holds the same regardless of the patch size that the model is trained on. Moreover, our parametric model is able to scale to large dataset without bringing additional computations.

5.2 Image Super-Resolution

Image super-resolution is a task to produce a high resolution image given a single low resolution one. Whereas previous MRF-based models [2,55] work reasonably,

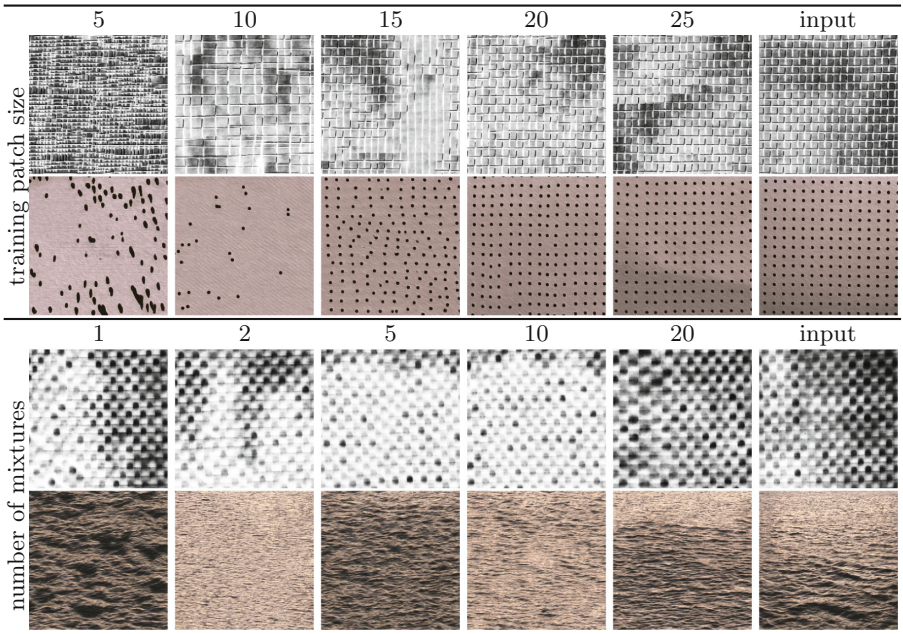

Fig. 6. Texture synthesis by varying the patch size and the number of mixtures.

the quality of their products is inferior to the state-of-the-art models based on deep learning [52,54]. With deep MRF, we wish to close the gap.

Unlike in texture synthesis, the generation of this task is driven by a low-resolution image. To incorporate this information, we introduce additional connections between the hidden states and corresponding pixels of the low-resolution image, as shown in Fig. 7. It is noteworthy that we just input *a single pixel* (instead of a *patch*) at each site, and in this way, we can test whether the model can propagate information across the spatial domain. As the task is deterministic, we use a GMM with a single component and fix its variance. In the testing stage, we output the mean of the Gaussian component at each location as the

Table 1. PSNR (dB) on Set5 dataset with upscale factor 2,3,4

Images	2× upscale				3× upscale				4× upscale			
	Bicubic	CNN	SE	Ours	Bicubic	CNN	SE	Ours	Bicubic	CNN	SE	Ours
Baby	37.07	38.30	**38.48**	38.31	33.91	35.01	**35.22**	35.15	31.78	32.98	**33.14**	32.94
Bird	36.81	40.40	**40.50**	40.36	32.58	34.91	35.58	**36.14**	30.18	31.98	**32.54**	32.49
Butterfly	27.43	32.20	31.86	**32.74**	24.04	27.58	26.86	**29.09**	22.10	25.07	24.09	**25.78**
Head	34.86	35.64	35.69	**35.70**	32.88	33.55	**33.76**	33.63	31.59	32.19	**32.52**	32.41
Women	32.14	34.94	**35.33**	34.84	28.56	30.92	31.36	**31.69**	26.46	28.21	28.92	**28.97**
Average	33.66	36.34	36.37	**36.38**	31.92	32.30	32.56	**33.14**	28.42	30.09	30.24	**30.52**

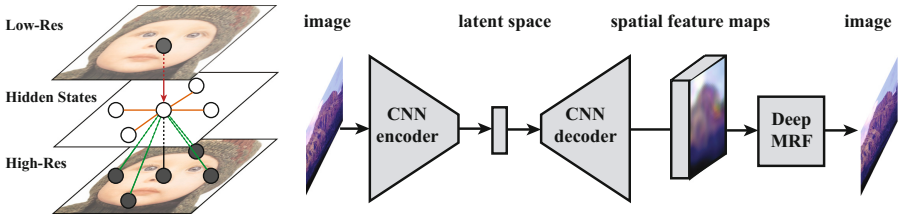

Fig. 7. Adapting deep MRFs to specific applications. Image super-resolution: the hidden state receives an additional connection from the low-resolution pixel. Image synthesis: deep MRF renders the final image from a spatial feature map, which is jointly learned by a variational auto-encoder.

Table 2. PSNR (dB) on various dataset with upscale factor 3

Dataset	Bicubic	A+ [51]	CNN [52]	SE [53]	CSCN [54]	Ours
Set5	30.39	32.59	32.30	32.56	33.10	**33.14**
Set14	27.54	29.13	29.00	29.16	**29.41**	29.38
BSD100	27.22	28.18	28.20	28.20	28.50	**28.54**

inferred high-resolution pixel. This approach is very generic – the model is not specifically tuned for the task and no pre- and post-processing steps are needed.

We train our model on a widely used super-resolution dataset [56] which contains 91 images, and test it on Set5, Set14, and BSD100 [57]. The training is on patches of size 16×16 and *rmsprop* with momentum 0.95 is used. We use PSNR for quantitative evaluation. Following previous work, we only consider the luminance channel in the *YCrCb* color space. The two chrominance channels are upsampled with bicubic interpolation.

As shown in Tables 1 and 2, our approach outperforms the CNN-based baseline [52] and compares favorably with the state-of-the-art methods dedicated to this task [53,54]. One possible explanation for the success is that our model not only learns the mapping, but also learns the image statistics for high resolution images. The training procedure which unrolls the RNN into thousands of steps that share parameters also reduces the risk of overfitting. The results also demonstrate the particular strength of our model in handling large upscaling factors and difficult images. Figure 8 shows several examples visually.

5.3 Natural Image Synthesis

Images can be roughly considered as a composition of textures with the guidance of scene and object structures. In this task, we move beyond the synthesis of homogeneous textures, and try to generate natural images with structural guidance.

While our model excels in capturing spatial dependencies, learning weak dependencies across the entire image is both computationally infeasible and

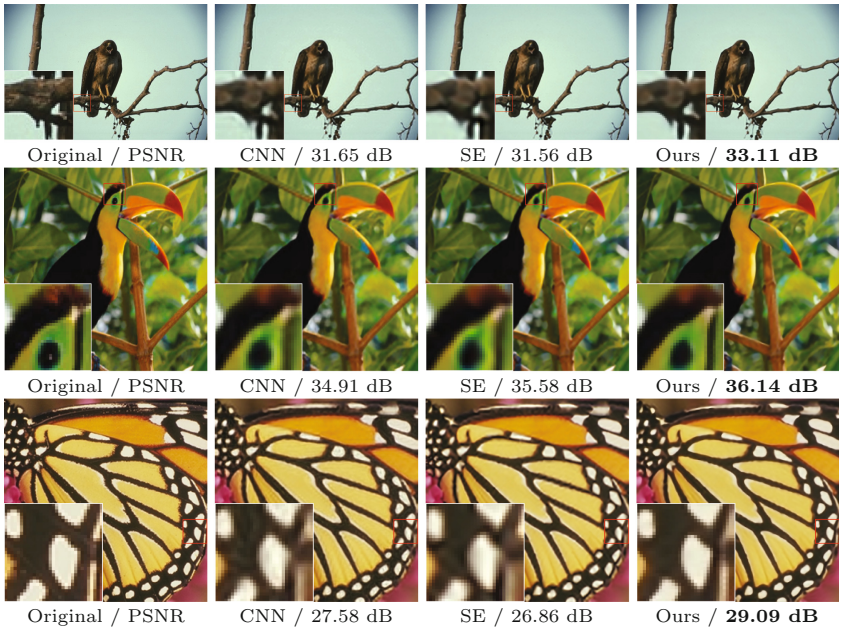

Fig. 8. Image super resolution results from Set 5 with upscaling factor 3.

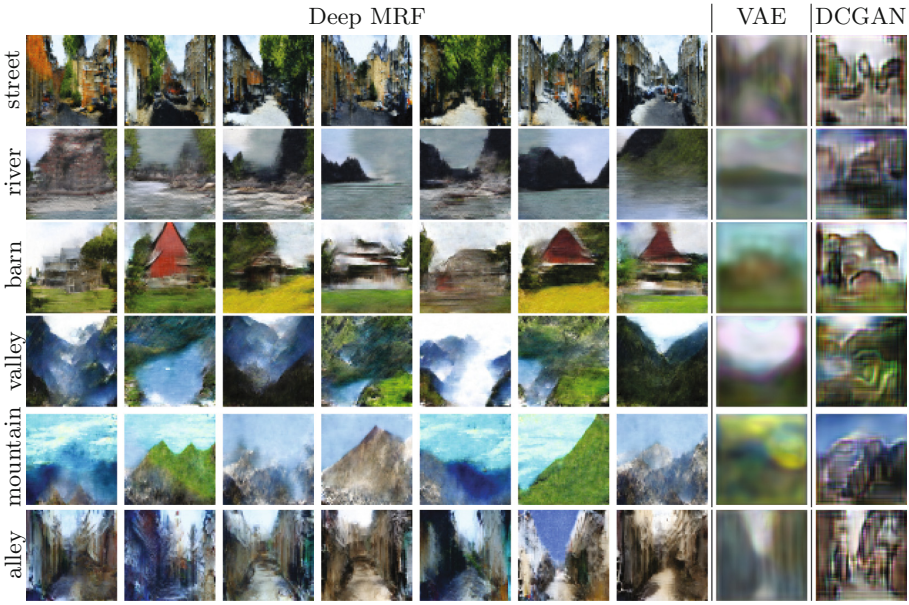

Fig. 9. Image synthesis results.

analytically inefficient. Instead, we adopt a global model to capture the overall structure and use it to provide contextual guidance to MRF. Specifically, we incorporate the *variational auto-encoder (VAE)* [10] for this purpose – VAE generates feature maps at each location and our model uses that feature to render the final image (see Fig. 7). Such features may contain information of scene layouts, objects, and texture categories.

We train the joint model end-to-end from scratch. During each iteration, the VAE first encodes the image into a latent vector, then decodes it to a feature map with the same size of the input image. We then connect this feature map to the latent states of the deep MRF. The total loss is defined as the addition of gaussian mixtures at image space and KL divergence at high-level VAE latent space. For training, we randomly extracts patches from the feature map. The gradients from the deep MRF back to the VAE thus only cover the patches being extracted. During testing, VAE randomly samples from the latent space and decodes it to generate the global feature maps. The output pixels are sampled from the GMM with 10 mixtures along the coupled acyclic graph.

We work on the MSRC [58] and SUN database [59] and select some scene categories with rich natural textures, such as *Mountains* and *Valleys*. Each category contains about a hundred images. As we will see, our approach generalizes much better than the data-hungry CNN approaches. We train the model on images of size 64×64 with a batch size of 4. For each image, we extract 16 patches of size 15×15 for training. Figure 9 shows several images generated from our models, in comparison with those obtained from the baselines, namely raw VAE [10] and DCGAN [60]. The CNN architecture is shared for all methods described in the DCGAN paper [60] to ensure fair comparison. We can see our model successfully captures a variety of local patterns, such as water, clouds, wall and trees. The global appearance also looks coherent, real and dynamic. The state-of-the-art CNN based models, which focuses too much on global structures, often yield sub-optimal local effects.

6 Conclusions

We present a new class of MRF model whose potential functions are expressed by powerful fully-connected neurons. Through theoretical analysis, we draw close connections between probabilistic deep MRFs and end-to-end RNNs. To tackle the difficulty of inference in cyclic graphs, we derive a new framework that decouples a cyclic graph with multiple coupled acyclic passes. Experimental results show state-of-the-art results on a variety of low-level vision problems, which demonstrate the strong capability of MRFs with expressive potential functions.

Acknowledgments. This work is supported by the Big Data Collaboration Research grant (CUHK Agreement No. TS1610626) and the Early Career Scheme (ECS) grant (No: 24204215). We also thank Aditya Khosla who participated in a discussion that is partly related to this work.

References

1. Portilla, J., Strela, V., Wainwright, M.J., Simoncelli, E.P.: Image denoising using scale mixtures of Gaussians in the wavelet domain. IEEE Trans. Image Process. **12**(11), 1338–1351 (2003)
2. Freeman, W.T., Pasztor, E.C., Carmichael, O.T.: Learning low-level vision. Int. J. Comput. Vis. **40**(1), 25–47 (2000)
3. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co., pp. 417–424 (2000)
4. McMillan, L., Bishop, G.: Plenoptic modeling: an image-based rendering system. In: Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques, pp. 39–46. ACM (1995)
5. Huang, J., Mumford, D.: Statistics of natural images and models. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1. IEEE (1999)
6. Turk, M.A., Pentland, A.P.: Face recognition using eigenfaces. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 1991, pp. 586–591. IEEE (1991)
7. Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T.S., Yan, S.: Sparse representation for computer vision and pattern recognition. Proc. IEEE **98**(6), 1031–1044 (2010)
8. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Comput. **14**(8), 1771–1800 (2002)
9. Werbos, P.J.: Backpropagation through time: what it does and how to do it. Proc. IEEE **78**(10), 1550–1560 (1990)
10. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
11. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
12. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a laplacian pyramid of adversarial networks. In: Advances in Neural Information Processing Systems, pp. 1486–1494 (2015)
13. Efros, A., Leung, T.K., et al.: Texture synthesis by non-parametric sampling. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1033–1038. IEEE (1999)
14. Wei, L.Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co., pp. 479–488 (2000)
15. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 327–340. ACM (2001)
16. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 341–346. ACM (2001)
17. Hays, J., Efros, A.A.: Scene completion using millions of photographs. ACM Trans. Graph. (TOG) **26**(3) (2007). Article no. 4

18. Lalonde, J.F., Hoiem, D., Efros, A.A., Rother, C., Winn, J., Criminisi, A.: Photo clip art. ACM Trans. Graph. (TOG) **26**(3) (2007). Article no. 3
19. Cross, G.R., Jain, A.K.: Markov random field texture models. IEEE Trans. Pattern Anal. Mach. Intell. **5**(1), 25–39 (1983)
20. Boykov, Y.Y., Jolly, M.P.: Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In: Proceedings of Eighth IEEE International Conference on Computer Vision, ICCV 2001, vol. 1, pp. 105–112. IEEE (2001)
21. He, X., Zemel, R.S., Carreira-Perpiñán, M.: Multiscale conditional random fields for image labeling. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, vol. 2, p. II-695. IEEE (2004)
22. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell. **6**(6), 721–741 (1984)
23. Ising, E.: Beitrag zur theorie des ferromagnetismus. Zeitschrift für Physik A Hadrons and Nuclei **31**(1), 253–258 (1925)
24. Rue, H., Held, L.: Gaussian Markov Random Fields: Theory and Applications. CRC Press, London (2005)
25. Zhu, S.C., Wu, Y., Mumford, D.: Filters, random fields and maximum entropy (frame): towards a unified theory for texture modeling. Int. J. Comput. Vis. **27**(2), 107–126 (1998)
26. Roth, S., Black, M.J.: Fields of experts: a framework for learning image priors. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, pp. 860–867. IEEE (2005)
27. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2625–2634 (2015)
28. Mnih, V., Heess, N., Graves, A., et al.: Recurrent models of visual attention. In: Advances in Neural Information Processing Systems, pp. 2204–2212 (2014)
29. Gregor, K., Danihelka, I., Graves, A., Wierstra, D.: Draw: a recurrent neural network for image generation. arXiv preprint arXiv:1502.04623 (2015)
30. Graves, A., Schmidhuber, J.: Offline handwriting recognition with multidimensional recurrent neural networks. In: Advances in Neural Information Processing Systems, pp. 545–552 (2009)
31. Graves, A., Fernandez, S., Schmidhuber, J.: Multi-dimensional recurrent neural networks. arXiv preprint arXiv:0705.2011 (2007)
32. Byeon, W., Breuel, T.M., Raue, F., Liwicki, M.: Scene labeling with LSTM recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3547–3555 (2015)
33. Bell, S., Zitnick, C.L., Bala, K., Girshick, R.: Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks. arXiv preprint arXiv:1512.04143 (2015)
34. Theis, L., Bethge, M.: Generative image modeling using spatial LSTMs. In: Advances in Neural Information Processing Systems, pp. 1918–1926 (2015)
35. Oord van den, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. arXiv preprint arXiv:1601.06759 (2016)
36. Rangarajan, A., Chellappa, R., Manjunath, B.: Markov random fields and neural networks with applications to early vision problems. Citeseer (1991)

37. Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., Torr, P.H.: Conditional random fields as recurrent neural networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1529–1537 (2015)
38. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv preprint arXiv:1412.7062 (2014)
39. Tompson, J.J., Jain, A., LeCun, Y., Bregler, C.: Joint training of a convolutional network and a graphical model for human pose estimation. In: Advances in Neural Information Processing Systems, pp. 1799–1807 (2014)
40. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
41. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo (2014)
42. Li, S.Z.: Markov Random Field Modeling in Image Analysis. Springer Science & Business Media, London (2009)
43. Salakhutdinov, R.R.: Learning in Markov random fields using tempered transitions. In: Advances in Neural Information Processing Systems, pp. 1598–1606 (2009)
44. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. **12**, 2121–2159 (2011)
45. Graves, A.: Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850 (2013)
46. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. **39**(1), 1–38 (1977)
47. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
48. Berglund, M., Raiko, T., Honkala, M., Kärkkäinen, L., Vetek, A., Karhunen, J.: Bidirectional recurrent neural networks as generative models-reconstructing gaps in time series. arXiv preprint arXiv:1504.01575 (2015)
49. Brodatz, P.: Textures: a photographic album for artists and designers, 1966. Images downloaded in July (2009)
50. MIT Media Lab: Vision Texture Database (2002). http://vismod.media.mit.edu
51. Timofte, R., De Smet, V., Van Gool, L.: A+: adjusted anchored neighborhood regression for fast super-resolution. In: Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS, vol. 9006, pp. 111–126. Springer, Heidelberg (2015)
52. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part IV. LNCS, vol. 8692, pp. 184–199. Springer, Heidelberg (2014)
53. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5197–5206. IEEE (2015)
54. Wang, Z., Liu, D., Yang, J., Han, W., Huang, T.: Deep networks for image super-resolution with sparse prior. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 370–378 (2015)
55. Freeman, W., Liu, C.: Markov random fields for super-resolution and texture synthesis. In: Advances in Markov Random Fields for Vision and Image Processing, vol. 1 (2011)
56. Bevilacqua, M., Roumy, A., Guillemot, C., Alberi-Morel, M.L.: Low-complexity single-image super-resolution based on nonnegative neighbor embedding, BMVA press (2012)

57. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of 8th International Conference on Computer Vision, vol. 2, pp. 416–423, July 2001
58. Shotton, J., Winn, J., Rother, C., Criminisi, A.: Textonboost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout, and context. Int. J. Comput. Vis. **81**(1), 2–23 (2009)
59. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: large-scale scene recognition from abbey to zoo. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3485–3492. IEEE (2010)
60. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)