2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Deep Structured Energy-Based Image Inpainting

Fazil Altinel*, Mete Ozay*, and Takayuki Okatani*†
*Graduate School of Information Sciences, Tohoku University, Sendai, Japan
†RIKEN Center for AIP, Tokyo, Japan
Email: {altinel, mozay, okatani}@vision.is.tohoku.ac.jp

***Abstract*—In this paper, we propose a structured image inpainting method employing an energy based model. In order to learn structural relationship between patterns observed in images and missing regions of the images, we employ an energy-based structured prediction method. The structural relationship is learned by minimizing an energy function which is defined by a simple convolutional neural network. The experimental results on various benchmark datasets show that our proposed method significantly outperforms the state-of-the-art methods which use Generative Adversarial Networks (GANs). We obtained 497.35 mean squared error (MSE) on the Olivetti face dataset compared to 833.0 MSE provided by the state-of-the-art method. Moreover, we obtained 28.4 dB peak signal to noise ratio (PSNR) on the SVHN dataset and 23.53 dB on the CelebA dataset, compared to 22.3 dB and 21.3 dB, provided by the state-of-the-art methods, respectively. The code is publicly available.[1]**

I. Introduction

In this work, we address an image inpainting problem, which is to recover missing regions of an image, as shown in Fig.1. The key to the problem is how to model structural relationship between local regions of natural images, and then use it for estimating pixel intensities of the missing regions. Indeed, this has been a major concern in previous studies of image inpainting, and therefore it is often formulated as a structured prediction problem [1], [2]. In general, structured prediction problems are formulated as energy minimization; an energy function is incorporated to model structural relationship between unknowns (e.g., pixel intensities in the case of image inpainting), and the unknowns are estimated by minimizing it.

Recently, deep networks were applied to image inpainting, and have achieved great success [3]–[5]. In these works, deep networks are trained using training samples so that they can estimate pixel intensities of missing regions of an input image; the computation is performed in a feedforward manner from an input image to an output inpainted image. In the training step, generative adversarial networks (GANs) are popularly used. In GANs which do not employ energy functions, the structural relationship between image pixels is implicitly represented internally inside the deep networks, which is obtained through learning (and partly modeled by the network architectures). Likewise, similar deep-network-based approaches have been applied with success to various problems which were formerly treated as structured prediction problems, such as human pose estimation, image super-resolution, etc.

In parallel to this trend, Belanger et al. [6] proposed to use deep networks in the framework of structured prediction,

[1]https://github.com/cvlab-tohoku/DSEBImageInpainting.

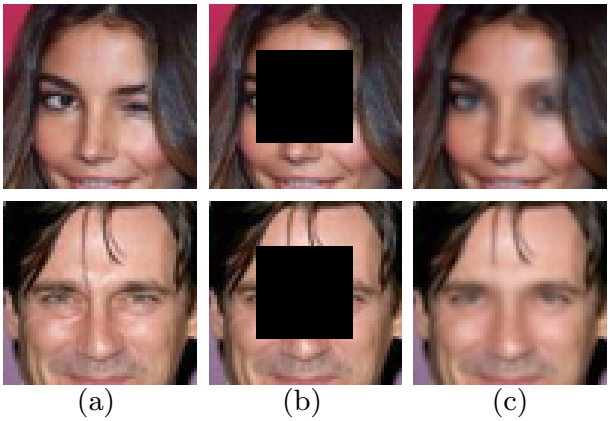

Fig. 1. An illustration of image inpainting results on face images. Given images with missing region (b), our method generates visually realistic and consistent inpainted images (c). (a) Ground truth images, (b) occluded input images, (c) inpainting results by our method.

where deep networks are used to define energy functions. As compared with previous approaches such as manually designed energy functions or training of shallow models (e.g., conditional random fields (CRFs)), their approach is expected to improve the representation power of energy functions as well as their accuracy in modeling structural relationship through learning from data. The structural relationship thus learned is still implicitly represented internally in deep networks. However, the performance of this method to various structured prediction problems is not so clear as of now; for instance, it remains unclear which approach is better for individual problems between the energy-based and the non-energy-based methods, both of which use deep networks. This must and can only be clarified through evaluations by its applications to various tasks.

With these in mind, in this study, we apply the energy-based method that uses deep networks to image inpainting, while there is no study in the literature to the authors knowledge. As shown in Fig. 1, we aim to learn the structural relationship between the unmasked and masked regions in a natural image. For this purpose, we propose a two-path CNN having inter-layer connections between the paths, which takes an input image with missing regions and any image (which is to be optimized in the inference phase) as the inputs of the two paths, respectively. It outputs energy for the input image pair from its final layer. In the training phase, we train this CNN

using a dataset of a pair of an original image and its masked version. The trained CNN provides an energy function that models *deep structural relationship* between unmasked and masked regions for different images. In the inference phase, given an image with masked regions, we minimize the energy function realized by the CNN, yielding an inpainted image in which masked regions are filled with estimated pixels (Section III).

We tested the proposed method on benchmark image inpainting datasets [7]–[9]. We evaluate the proposed method qualitatively and quantitatively, and show its success compared to the state-of-the-art methods [3], [4] in Section IV. The results show that our proposed method successfully generates both realistic and accurate inpainted images for given masked images. Our contributions are summarized as follows:

- We propose a simple yet efficient deep structured image inpainting method. In order to learn the structural relationship between image pixels in the proposed method, we suggest an energy based model which employs an energy function defined by a CNN.
- We demonstrate results of our method on various benchmark image inpainting datasets to show the effectiveness of our method. Our proposed method obtains state-of-the-art performance, and generates more realistic and accurate inpainted images compared to the baseline methods.

II. Related Work

Recently, deep learning has been employed for image inpainting and provided remarkable results [1], [10], [11]. The methods which use Generative Adversarial Networks (GANs) [5] achieved promising results on benchmark image inpainting tasks [3], [4], [12].

Pathak et al. [3] proposed an encoder-decoder network for image inpainting called Context Encoders (CEs). CEs are trained by minimizing a function of ℓ_2 loss and adversarial loss [5] to inpaint occluded image regions. Although CEs are able to generate promising inpainting results, they make use of structures of occluded regions during training but not during inference. Thus, inpainting results of CEs are sometimes visually unrealistic.

Yeh et al. [4] proposed a semantic image inpainting framework which includes contextual and perceptual losses. Their framework leverages a pre-trained Deep Convolutional GAN (DCGAN) [13], and aims to find the closest mapping in the latent space. However, inpainting results of their framework usually have differences in color along mask boundaries. Therefore, some post-processing methods, such as Poisson blending [14], are used to eliminate color differences after inpainted images are generated by the framework. Moreover, their method requires many iterations to find the closest mapping in the latent space during testing phase. On the other hand, our method can generate realistic and visually consistent contents, and requires no post-processing.

Iizuka et al. [12] proposed an approach for image completion using deep neural networks (DNNs). Their method is *non-blind inpainting*, that is, it requires not only an input image but a mask indicating regions to be inpainted. The above two methods [3], [4] including ours perform blind inpainting where no mask is necessary.

In addition, DNNs have been used for structured prediction due to its generative power [1], [2], [5], [15]–[18]. Predicting structured outputs using energy based learning was analyzed in [19]. Belanger et al. [6] proposed a structured prediction energy network (SPEN) in which a DNN is exploited to define an energy function, and predictions are obtained by minimizing the energy function using gradient descent. Belanger et al. [20] introduced an end-to-end learning method for SPENs, where an energy function is minimized by back-propagation using gradient-based prediction.

Amos et al. [21] proposed input convex neural networks (ICNNs) which share similar architectural properties compared to SPENs. ICNNs add constraints to the parameters of SPENs, such that their energy function is convex with respect to some parameters of the energy function and the optimization can be performed globally. Our architecture is similar to that of SPENs and ICNNs. Unlike SPENs, which only consider multi-label classification problems, we address image inpainting problems. Employment of end-to-end learning methods enables SPENs to handle more complex tasks such as depth image denoising.

We employ a specific CNN which includes connections from input layer to hidden layers in order to achieve realistic image inpainting results. Such connections have been recently discussed in deep residual networks [22] and densely connected convolutional networks [23]. ICNNs employ such connections in order to restrict the energy function to be a convex function with respect to some parameters of the energy function. However, convexity gives a strong restriction on the expressivity of the energy function. In our experiments, we show that convexity constraints of ICNNs hinder the networks from generating visually better image inpainting results, and thus our method performs better than ICNNs.

III. Proposed Method

The proposed framework is illustrated in Fig. 2. The design of the network representing an energy function, learning procedures, and testing phase in our proposed framework are described in Section III-A, III-B, and III-C, respectively.

A. CNN Representing an Energy Function

We use a CNN to represent an energy function $E_x(\hat{y};\Theta)$, where x is an input image with missing regions, $\hat{y}$ is an estimate of the true image y (i.e., the inpainted version of x), and Θ is the parameters (weights) of the CNN. Figure 3 shows its architecture. It contains two paths called input path Π_I and output path Π_O, each of which has L-layers. The image x is fed to Π_I, and the image $\hat{y}$ is fed to Π_O. Π_I has parameters $\theta_{\Pi_I} = \{W^{(u)}_{l,k}, b^{(u)}_{l,k} : k = 1, 2, \ldots, K\}^L_{l=1}$ and Π_O has parameters $\theta_{\Pi_O} = \{W^{(v)}_{l,k}, W^{(u)}_{l,k}, W^{(z)}_{l,k}, b^{(v)}_{l,k} : k = 1, 2, \ldots, K\}^L_{l=1}$. They comprise all the trainable parameters of the CNN, i.e., $\Theta = \{\theta_{\Pi_I}, \theta_{\Pi_O}\}$.

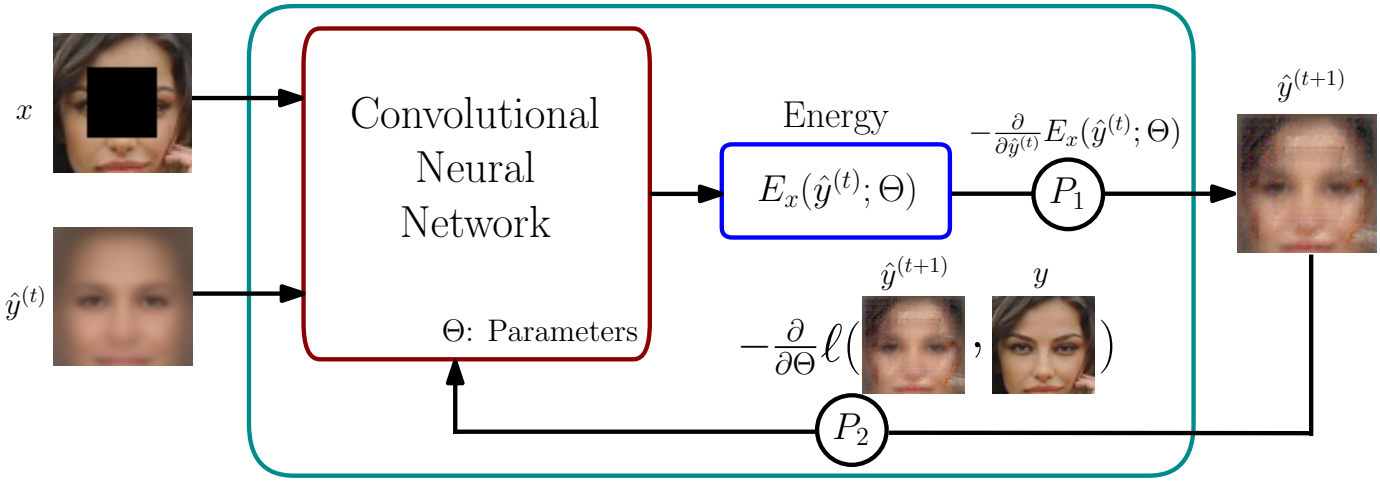

Fig. 2. An overview of the proposed method. The network takes an occluded image x and an estimate $\hat{y}^{(t)}$ of the inpainted, true image as inputs. The operation P_1 performs gradient descent to compute the minimizer $\hat{y}$ to $E_x(\hat{y};\Theta)$ for a given Θ. The number of iterations is set to 1 for simplicity of illustration. The operation P_2 updates the network parameter Θ by gradient descent to minimize the loss function $\ell(\hat{y}^{(t+1)}, y)$. For learning the network parameter Θ, P_1 and P_2 are alternately performed, whereas for estimating the inpainted image, only P_1 is performed using learned Θ.

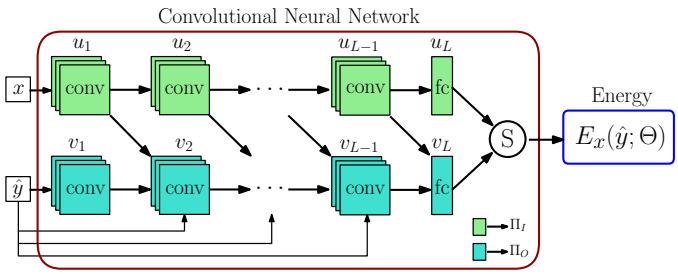

Fig. 3. The CNN representing the energy function. It computes a scalar energy $E_x(\hat{y};\Theta)$, given an input image x with missing regions and an estimate $\hat{y}$ of the true inpainted image.

Input path Π_I: We compute the k^{th} feature map at the $(l+1)^{st}$ layer of the input path Π_I by

$$u_{l+1}^k = \sigma(W_{l,k}^{(u)} * u_l^k + b_{l,k}^{(u)}), \tag{1}$$

where $*$ denotes the convolution operation; $\sigma(\cdot)$ is an activation function; $W_{l,k}^{(u)}$ and $b_{l,k}^{(u)}$ denotes the weight matrix and the bias vector applied to the k^{th} feature map u_l^k, ($k = 1, 2, \ldots, K$ and $l = 1, 2, \ldots, L$). At the input layer ($l = 0$), we set $u_0^k = x$. Following the convolutional layers, fully connected layers are used in Π_I. In Fig. 3, $u_l = [u_l^1, u_l^2, \ldots, u_l^K]$ denotes a tensor of the feature maps obtained at the l^{th} layer of Π_I.

Output path Π_O: As shown in Fig. 3, there are connections from Π_I to Π_O. Specifically, each layer $l(\geq 2)$ of this path has input connections from the lower layer output of Π_I (i.e., u_{l-1}) in addition to the lower layer output (i.e., v_{l-1}). Furthermore, each layer except the last fully connected (FC) layer has a connection from the input $\hat{y}$, which is rescaled to fit to the size of its input feature map. Thus, the k^{th} feature map v_{l+1}^k is computed by

$$v_{l+1}^k = \sigma(W_{l,k}^{(v)} * v_l^k + W_{l,k}^{(u)} * u_l^k + W_{l,k}^{(z)} * z_l^k + b_{l,k}^{(v)}), \tag{2}$$

where z_l^k is the rescaled version of $\hat{y}$ at the l^{th} layer; $W_{l,k}^{(v)}$ and $W_{l,k}^{(z)}$ are the weights applied on the feature maps v_l^k and z_l^k, respectively ($k = 1, 2, \ldots, K$ and $l = 1, 2, \ldots, L$). At the input layer ($l = 0$) of the output path Π_O, we set $W_{0,k}^{(z)} = 0$, $W_{0,k}^{(u)} = 0$, and $v_0^k = \hat{y}$.

In order to compute the scalar value of the energy function $E_x(\hat{y};\Theta)$ for the inputs x and $\hat{y}$ of the CNN, an operation (denoted by S) is employed following the top FC layers of the input path Π_I and the output path Π_O (see Fig. 3). The S-operation performs element-wise addition on the results obtained from the FC layers. Then, the output of the S-operation is reshaped to a scalar value of the energy.

B. Learning the Energy Function

We learn the energy function $E_x(\hat{y};\Theta)$ (i.e., its parameter Θ) using N pairs of an input image x and its true image y. We denote the set of input images and that of the true images by $\mathcal{X} = \{x_i\}_{i=1}^N$ and $\mathcal{Y} = \{y_i\}_{i=1}^N$, respectively. In order to determine Θ, we consider the following constrained minimization problem:

$$\min_{\Theta} \ell(\hat{y}, y) \quad \text{subject to} \quad \hat{y} = \underset{\hat{y}}{\operatorname{argmin}}\, E_x(\hat{y};\Theta). \tag{3}$$

The underlying idea [6] is that we wish to obtain Θ such that the minimizer $\hat{y}$ to $E_x(\hat{y};\Theta)$ for a given x and Θ provides the most similar image $\hat{y}$ to the ground truth y for any pair of x and y. We use the ℓ_1 distance between $\hat{y}$ and y for $l(\hat{y}, y)$:

$$\ell(\hat{y}, y) = \|\hat{y} - y\|_1. \tag{4}$$

The constrained minimization (3) can be converted to the following unconstrained minimization, if we can write the minimizer $\hat{y}$ to $E_x(\hat{y};\Theta)$ for a given Θ as $\hat{y}(\Theta)$:

$$\min_{\Theta} l(\hat{y}(\Theta), y). \tag{5}$$

Algorithm 1: Our proposed training algorithm. Learning rate used for optimization of energy $E_x(\hat{y};\Theta)$ in T iterations is denoted by α, and learning rate used for optimization of parameters Θ is denoted by λ.

Input: $\{(x_i, y_i)\}_{i=1}^{N}$: Training set of image pairs.
Input: Θ : Parameters of the network.
1 $\hat{y}^{(0)} \leftarrow \frac{1}{N}\sum_{i=1}^{N} y_i$;
2 Initialize the parameters $\Theta^{(0)}$;
3 $E_x^{(0)} \leftarrow E_x(\hat{y}^{(0)};\Theta^{(0)})$;
4 **for** $m \leftarrow 0$ **to** M **do**
5 Randomly choose an image pair $(x, y) \in \{(x_i, y_i)\}_{i=1}^{N}$ from the training set;
6 **for** $t \leftarrow 0$ **to** T **do**
7 $\hat{y}^{(t+1)} \leftarrow \hat{y}^{(t)} - \alpha\frac{\partial}{\partial \hat{y}^{(t)}} E_x^{(t)}$;
8 $E_x^{(t+1)} \leftarrow E_x(\hat{y}^{(t+1)};\Theta^{(m)})$;
9 **end**
10 $\Theta^{(m+1)} \leftarrow \Theta^{(m)} - \lambda\frac{\partial}{\partial \Theta^{(m)}}\ell(\hat{y}^{(T)}, y)$;
11 **end**

Thus, for a training sample (x_i,y_i), we first minimize $E_x(\hat{y};\Theta)$ for a fixed Θ with respect to $\hat{y}$ using gradient descent and then update Θ in the direction to minimize $l(\hat{y}(\Theta), y)$ using gradient descent. We iterate this pair of computation of $\hat{y}(\Theta)$ and update of Θ until convergence using training samples $\mathcal{X}$ and $\mathcal{Y}$. In Fig. 2, the computation of $\hat{y}(\Theta)$ is denoted by P_1 and the updating of Θ is denoted by P_2.

The overall algorithm is given in Algorithm 1. For simplicity of explanation, a batch of size 1 is used here. The initial value $\hat{y}^{(0)}$ of $\hat{y}$ is set to be the mean image of the ground truth images belonging to the set $\mathcal{Y}$. Then, the initial energy can be computed using an occluded image x, the image $\hat{y}^{(0)}$ and initial parameters $\Theta^{(0)}$ of the network. After the maximum number T of iterations of the update of $\hat{y}$ with a fixed $\Theta = \Theta^{(m)}$ is achieved, the resulting $\hat{y}(\Theta)$ is used to update $\Theta^{(m)}$ to minimize the loss function $\ell(\hat{y}^{(T)}, y)$. The learning phase is completed after M iterations.

C. Inference for a Novel Image x

Once the energy is learned, we can obtain an inpainted image for a novel input image x with missing regions. This is done by minimizing the energy $E_x(\hat{y};\Theta)$ with respect to $\hat{y}$, while fixing Θ and x. In order to perform this minimization, we first initialize $\hat{y}$ with a mean image as in the first step of Algorithm 1. Then, we iteratively update $\hat{y}$ for $t = 1, 2, \ldots, T$ according to line 7 of Algorithm 1. Our estimate is $\hat{y}^{(T)}$.

IV. Experimental Analyses

A. Implementation Details

For implementation of the network in the proposed method, we used a simple CNN architecture. Our architecture consists of three convolutional and one fully connected layers in all our experiments. Rectified Linear Unit (ReLU) is employed as the activation function. We used learning rate α of 0.01 and momentum of 0.9 for energy update, and learning rate λ of 0.001 for parameter update. Parameters were updated using the ADAM optimization algorithm [24]. We implemented our proposed method using the TensorFlow framework. We used the same experimental settings for training and testing. Peak Signal to Noise Ratio (PSNR) metric was employed to quantitatively evaluate the inpainting results by

$$PSNR = 10 \cdot \log_{10} \frac{255^2}{\varepsilon}, \tag{6}$$

where ε is the mean squared error between the inpainted image and the ground truth image. PSNR results were averaged for samples belonging to the test set, and an average PSNR value was given for each experiment. Implementation details, hyperparameters and the code were made publicly available at https://github.com/cvlab-tohoku/DSEBImageInpainting.

B. Datasets

We evaluate the proposed method using four datasets: The Olivetti face dataset [8], the MNIST [25], the Street View House Numbers (SVHN) [9], and the Large-scale CelebFaces Attributes (CelebA) [7]. The Olivetti face dataset [8] contains 400 grayscale face images of size 64×64, and 50 images of the dataset were used for testing. The MNIST dataset [25] contains 70,000 grayscale images of size 28×28. We used 10,000 images of the MNIST dataset for testing, and all images were resized to 64×64. The SVHN dataset [9] consists of 99,289 RGB color images. We used 73,257 images for training and 26,032 images for testing. The images in the training and testing sets were resized to 64×64. The CelebA dataset [7] contains 202,599 RGB color face images. We used 2000 images for testing. All images belonging to the CelebA dataset were cropped to 64×64 at the center in order to extract regions that cover only faces. We performed the inpainting tests using two types of block occlusions:

- Center block occlusion,
- Half block occlusion.

For center block occlusion tests, we created masks which cover approximately 25% of the images. We located the masks at the center of images for the center block occlusion tests. Moreover, we performed half block occlusion tests on the Olivetti face dataset creating only left half occlusion masks in order to compare with the existing methods. Center block occlusion tests were performed using all datasets except the Olivetti face dataset. Half block occlusion tests were performed using only the Olivetti face dataset.

C. Analyses using Grayscale Images

We first provide test results using grayscale images which belong to the Olivetti face dataset in our analyses. The first three columns of Fig. 4 depict the test results using the Olivetti face dataset. The results obtained using our proposed method for the half block occlusion inpainting task are shown in the last row of the figure. The results show that our proposed method generates images which are visually realistic and similar to ground truth images.

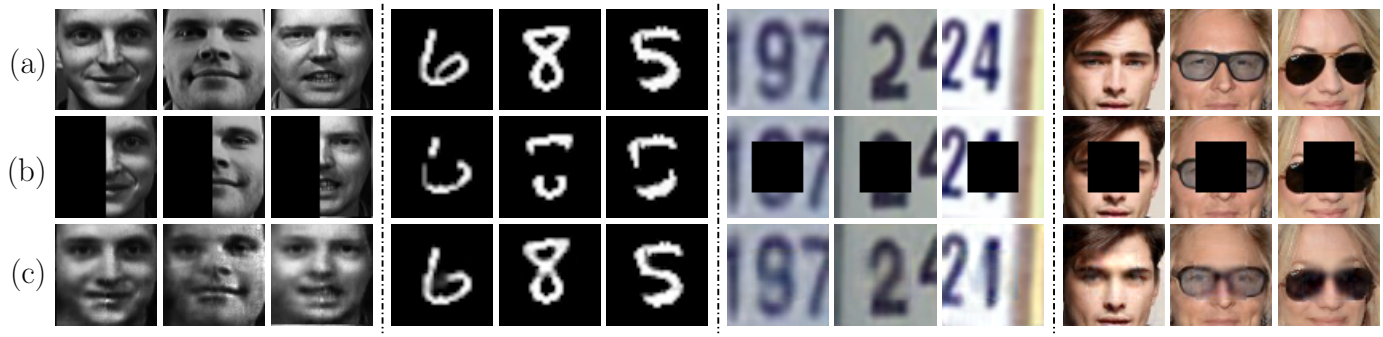

Fig. 4. Examples of results obtained in image inpainting experiments using the test split of the Olivetti face, the MNIST, the SVHN, and the CelebA datasets, respectively. (a) Ground truth images, (b) occluded input images, and (c) our inpainting results.

We exploited the same training and test splits, and minimized mean squared error (MMSE) in order to directly compare our results with the baseline methods [21], [26]. Table I shows the test MMSE results for our method and the state-of-the-art methods. As shown in the Table I, our proposed method outperforms the baseline methods using the Olivetti face dataset with a remarkable performance boost.

TABLE I
MEAN SQUARED ERRORS COMPUTED FOR THE HALF BLOCK OCCLUSION INPAINTING TASK USING THE TEST SPLIT OF THE OLIVETTI FACE DATASET.

Method	Mean Squared Error
Sum Product Networks [26]	942.0
Input Convex Neural Networks (ICNNs) [21]	833.0
Our method	**497.35**

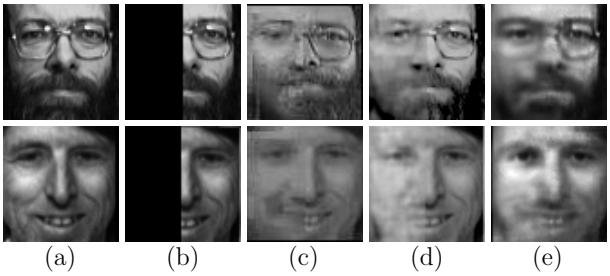

Fig. 5. Comparison of test results for the half block occlusion inpainting task using the Olivetti face dataset. (a) Ground truth images, (b) occluded input images, (c) Sum Product Networks [26], (d) ICNNs [21], (e) our method.

Fig. 5 shows a comparison of results obtained using the proposed method and the baseline methods [21], [26] for the half block occlusion inpainting task on the Olivetti face dataset. The last column shows our results in the figure. The results show that our method generates images which are visually more similar to the ground truth images compared to the baseline methods. The second three columns of Fig. 4 depict our results obtained for the center block occlusion inpainting task using the test split of the MNIST dataset. The (c) column of the figure shows our inpainting results.

D. Analyses using Color Images

In this subsection, we compare our results with the state-of-the-art Context Encoders [3] and Semantic Image Inpainting [4] methods. We used the same training and test splits, and masks that were used for analysis of the state-of-the-art methods [3], [4] for a fair comparison.

In the third three columns of Fig. 4, we show examples of our results obtained for the center block occlusion inpainting task using the SVHN dataset. The last row of the figure shows our inpainting results. The results show that our method generates correct digits even if larger parts of digits are occluded by masks.

TABLE II
THE RESULTS OBTAINED FOR THE CENTER BLOCK OCCLUSION INPAINTING TASK USING THE TEST SPLIT OF THE SVHN DATASET.

Method	PSNR (dB)
Context Encoders (CEs) [3]	22.3
Semantic Image Inpainting [4]	19.0
Our method	**28.4**

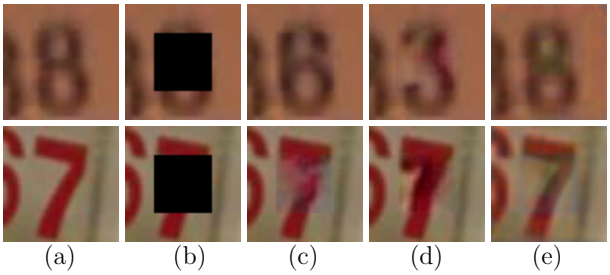

Fig. 6. Comparison of test results for the center block occlusion inpainting task using the SVHN dataset. (a) Ground truth images, (b) occluded input images, (c) CEs [3], (d) Semantic Image Inpainting [4], (e) our method.

We compare PSNR results on the SVHN dataset for the center block occlusion inpainting task obtained using Context Encoders [3], Semantic Image Inpainting [4], and our method in Table II. The ground truth images belonging to the test set of the SVHN dataset are used as reference to calculate the PSNR values. In the experiments, our method obtained the highest PSNR value providing 6.1 dB more than the PSNR provided by state-of-the-art method. The results also show that

our method generates visually more similar images than the state-of-the-art methods.

Fig. 6 shows samples of the results obtained for the center block occlusion inpainting task using the SVHN dataset. The results indicate that our method generates visually more accurate and pleasing results compared to the state-of-the-art methods. The last three columns of Fig. 4 show examples of results obtained for the center block occlusion inpainting task using the CelebA dataset. The last row of the figure shows our inpainting results. The results demonstrate that images generated by our method are similar to the ground truth images and visually realistic.

In Table III, we compare the PSNR results obtained using the CelebA dataset for the center block occlusion inpainting task. The ground truth images belonging to the testing set of the CelebA dataset were used as reference to calculate the PSNR values. These results indicate that the PSNR value of our method is 2.23 dB higher than that of the best of the state-of-the-art methods. Fig. 7 shows a comparison of the results for the center block occlusion inpainting task using the CelebA dataset. The last column of the table given in the figure shows our inpainting results. The results show that our method generates inpainted images which are visually more similar to the ground truth images.

TABLE III
THE RESULTS OBTAINED FOR THE CENTER BLOCK OCCLUSION INPAINTING TASK USING THE TEST SPLIT OF THE CELEBA DATASET.

Method	PSNR (dB)
Context Encoders (CEs) [3]	21.3
Semantic Image Inpainting [4]	19.4
Our method	**23.53**

(a) (b) (c) (d) (e)

Fig. 7. Comparison of test results for the center block occlusion inpainting task using the CelebA dataset. (a) Ground truth images, (b) occluded input images, (c) CEs [3], (d) Semantic Image Inpainting [4], (e) our method.

V. CONCLUSION

In this work, we proposed a deep structured energy-based image inpainting method. The method is based on an energy based model which employs the energy function defined by a CNN. The proposed method can successfully inpaint the occluded region in images with visually more clear and realistic content compared to the baseline methods. Qualitative and quantitative results show that our proposed method achieves the state-of-the-art inpainting results by learning structural relationship between the patterns observed in images and occluded region of the images. In future work, we plan to employ the proposed method for other computer vision and pattern recognition tasks such as image denoising, super-resolution, image classification and object detection.

ACKNOWLEDGEMENTS

This work was partly supported by CREST, JST Grant Number JPMJCR14D1, and the ImPACT Program Tough Robotics Challenge of the Council for Science, Technology, and Innovation (Cabinet Office, Government of Japan).

REFERENCES

[1] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *CVPR*, July 2017.
[2] L.-C. Chen, A. G. Schwing, A. L. Yuille, and R. Urtasun, "Learning deep structured models," in *ICML*, 2015.
[3] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros, "Context encoders: Feature learning by inpainting," in *CVPR*, 2016.
[4] R. A. Yeh, C. Chen, T. Y. Lim, S. A. G., M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *CVPR*, 2017.
[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.
[6] D. Belanger and A. McCallum, "Structured prediction energy networks," in *ICML*, 2016.
[7] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *ICCV*, 2015.
[8] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, Dec 1994, pp. 138–142.
[9] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS*, 2011.
[10] S. Liu, J. Pan, and M.-H. Yang, "Learning recursive filters for low-level vision via a hybrid neural network," in *ECCV*, 2016.
[11] R. Gao and K. Grauman, "On-demand learning for deep image restoration," in *ICCV*, 2017.
[12] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and Locally Consistent Image Completion," *ACM Transactions on Graphics (Proc. of SIGGRAPH 2017)*, vol. 36, no. 4, pp. 107:1–107:14, 2017.
[13] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *CoRR*, vol. abs/1511.06434, 2015.
[14] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, Jul. 2003.
[15] Y. Song, A. G. Schwing, R. S. Zemel, and R. Urtasun, "Training deep neural networks via direct loss minimization," in *ICML*, 2016.
[16] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," in *ICML*, 2015.
[17] D. J. Im, C. D. Kim, H. Jiang, and R. Memisevic, "Generating images with recurrent adversarial networks," *CoRR*, vol. abs/1602.05110, 2016.
[18] L. Theis and M. Bethge, "Generative image modeling using spatial lstms," in *NIPS*, 2015.
[19] Y. Lecun, S. Chopra, R. Hadsell, F. J. Huang, G. Bakir, T. Hofman, B. Schlkopf, A. Smola, and B. T. (eds, "A tutorial on energy-based learning," in *Predicting Structured Data*. MIT Press, 2006.
[20] D. Belanger, B. Yang, and A. McCallum, "End-to-end learning for structured prediction energy networks," in *ICML*, 2017.
[21] B. Amos, L. Xu, and J. Z. Kolter, "Input convex neural networks," in *ICML*, 2017.
[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
[23] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.
[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.
[25] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, 1998, pp. 2278–2324.
[26] H. Poon and P. Domingos, "Sum-product networks: A new deep architecture," in *ICCV Workshops*, 2011.