Image Inpainting via Generative Multi-column Convolutional Neural Networks

Yi Wang[1] **Xin Tao**[1,2] **Xiaojuan Qi**[1] **Xiaoyong Shen**[2] **Jiaya Jia**[1,2]
[1]The Chinese University of Hong Kong [2]YouTu Lab, Tencent
{yiwang, xtao, xjqi, leojia}@cse.cuhk.edu.hk goodshenxy@gmail.com

Abstract

In this paper, we propose a generative multi-column network for image inpainting. This network synthesizes different image components in a parallel manner within one stage. To better characterize global structures, we design a confidence-driven reconstruction loss while an implicit diversified MRF regularization is adopted to enhance local details. The multi-column network combined with the reconstruction and MRF loss propagates local and global information derived from context to the target inpainting regions. Extensive experiments on challenging street view, face, natural objects and scenes manifest that our method produces visual compelling results even without previously common post-processing.

1 Introduction

Image inpainting (also known as image completion) aims to estimate suitable pixel information to fill holes in images. It serves various applications such as object removal, image restoration, image denoising, to name a few. Though studied for many years, it remains an open and challenging problem since it is highly ill-posed. In order to generate realistic structures and textures, researchers resort to auxiliary information, from either surrounding image areas or external data.

A typical inpainting method exploits pixels under certain patch-wise similarity measures, addressing three important problems respectively to (1) extract suitable features to evaluate patch similarity; (2) find neighboring patches; and (3) to aggregate auxiliary information.

Features for Inpainting Suitable feature representations are very important to build connections between missing and known areas. In contrast to traditional patch-based methods using hand-crafted features, recent learning-based algorithms learn from data. From the model perspective, inpainting requires understanding of global information. For example, only by seeing the entire face, the system can determine eyes and nose position, as shown in top-right of Figure 1. On the other hand, pixel-level details are crucial for visual realism, *e.g.* texture of the skin/facade in Figure 1.

Recent CNN-based methods utilize encoder-decoder [18, 25, 24, 9, 26] networks to extract features and achieve impressive results. But there is still much room to consider features as a group of different components and combine both global semantics and local textures.

Reliable Similar Patches In both exemplar-based [7, 8, 4, 21, 10, 11, 2] and recent learning-based methods [18, 25, 24, 9, 26], explicit nearest-neighbor search is one of the key components for generation of realistic details. When missing areas originally contain structure different from context, the found neighbors may harm the generation process. Also, nearest-neighbor search during testing is also time-consuming. Unlike these solutions, we in this paper apply search only in the training phase with improved similarity measure. Testing is very efficient without the need of post-processing.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

Figure 1: Our inpainting results on building, face, and natural scene.

Spatial-variant Constraints Another important issue is that inpainting can take multiple candidates to fill holes. Thus, optimal results should be constrained in a spatially variant way – pixels close to area boundary are with few choices, while the central part can be less constrained. In fact, adversarial loss has already been used in recent methods [18, 25, 24, 9, 26] to learn multi-modality. Various weights are applied to loss [18, 25, 26] for boundary consistency. In this paper, we design a new spatial-variant weight to better handle this issue.

The overall framework is a *Generative Multi-column Convolutional Neural Network* (GMCNN) for image inpainting. The multi-column structure [3, 27, 1] is used since it can decompose images into components with different receptive fields and feature resolutions. Unlike multi-scale or coarse-to-fine strategies [24, 12] that use resized images, branches in our multi-column network directly use full-resolution input to characterize multi-scale feature representations regarding global and local information. A new *implicit diversified Markov random field* (ID-MRF) term is proposed and used in the training phase only. Rather than directly using the matched feature, which may lead to visual artifacts, we incorporate this term as regularization.

Additionally, we design a new *confidence-driven reconstruction loss* that constrains the generated content according to the spatial location. With all these improvements, the proposed method can produce high quality results considering boundary consistency, structure suitability and texture similarity, without any post-processing operations. Exemplar inpainting results are given in Figure 1.

2 Related Work

Exemplar-based Inpainting Among traditional methods, exemplar-based inpainting [7, 8, 4, 21, 10, 11, 2] copies and pastes matching patches in a pre-defined order. To preserve structure, patch priority computation specifies the patch filling order [4, 7, 8, 21]. With only low-level information, these methods cannot produce high-quality semantic structures that do not exist in examples, *e.g.*, faces and facades.

CNN Inpainting Since the seminal *context-encoder* work [18], deep CNNs have achieved significant progress. Pathak *et al.* proposed training an encoder-decoder CNN and minimizing pixel-wise reconstruction loss and adversarial loss. Built upon *context-encoder*, in [9], global and local discriminators helped improve the adversarial loss where a fully convolutional encoder-decoder structure was adopted. Besides encoder-decoder, U-net-like structure was also used [23].

Yang *et al.*[24] and Yu *et al.*[26] introduced coarse-to-fine CNNs for image inpainting. To generate more plausible and detailed texture, combination of CNN and Markov Random Field [24] was taken as the post-process to improve inpainting results from the coarse CNN. It is inevitably slow due to iterative MRF inference. Lately, Yu *et al.* conducted nearest neighbor search in deep feature space [26], which brings clearer texture to the filling regions compared with previous strategies of a single forward pass.

3 Our Method

Our inpainting system is trainable in an end-to-end fashion, which takes an image $\mathbf{X}$ and a binary region mask $\mathbf{M}$ (with value 0 for known pixels and 1 otherwise) as input. Unknown regions in image $\mathbf{X}$ are filled with zeros. It outputs a complete image $\hat{\mathbf{Y}}$. We detail our network design below.

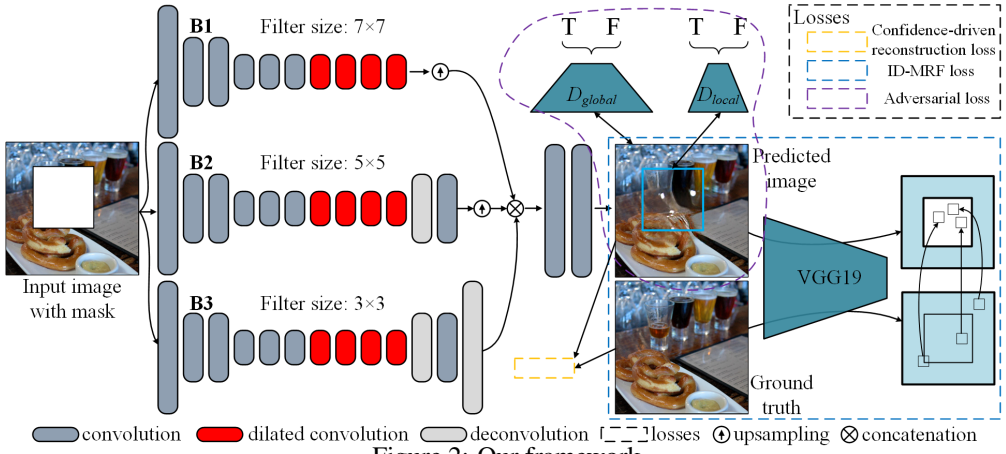


Figure 2: Our framework.

3.1 Network Structure

Our proposed *Generative Multi-column Convolutional Neural Network* (GMCNN) shown in Figure 2 consists of three sub-networks: a generator to produce results, global&local discriminators for adversarial training, and a pretrained VGG network [20] to calculate ID-MRF loss. In the testing phase, only the generator network is used.

The generator network consists of n ($n = 3$) parallel encoder-decoder branches to extract different levels of features from input $\mathbf{X}$ with mask $\mathbf{M}$, and a shared decoder module to transform deep features into natural image space $\hat{\mathbf{Y}}$. We choose various receptive fields and spatial resolutions for these branches as shown in Figure 2, which capture different levels of information. Branches are denoted as $\{f_i(\cdot)\}$ ($i \in \{1, 2, ..., n\}$), trained in a data driven manner to generate better feature components than handcrafted decomposition.

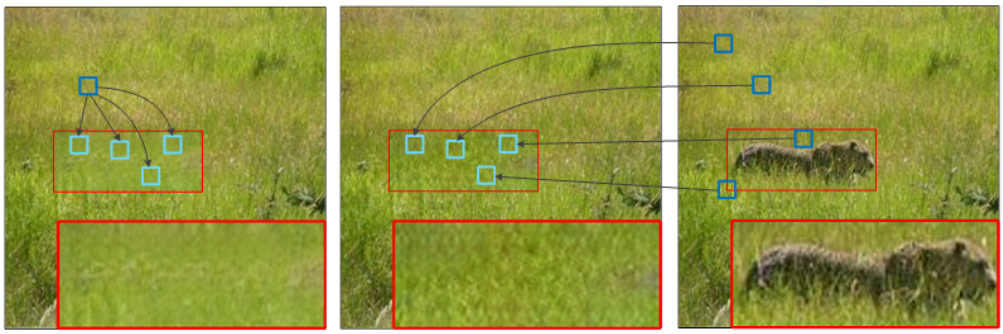
Then these components are up-sampled (bilinearly) to the original resolution and are concatenated into feature map F. We further transform features F into image space via shared decoding module with 2 convolutional layers, denoted as $d(\cdot)$. The output is $\hat{\mathbf{Y}} = d(F)$. Minimizing the difference between $\hat{\mathbf{Y}}$ and $\mathbf{Y}$ makes $\{f_i(\cdot)\}_{i=1,...,n}$ capture appropriate components in $\mathbf{X}$ for inpainting. $d(\cdot)$ further transforms such deep features to our desired result. Note that although $f_i(\cdot)$ seems independent of each other, they are mutually influenced during training due to $d(\cdot)$.

Analysis Our framework is by nature different from commonly used one-stream encoder-decoder structure and the coarse-to-fine architecture [24, 26, 12]. The encoder-decoder transforms the image into a common feature space with the same-size receptive field, ignoring the fact that inpainting involves different levels of representations. The multi-branch encoders in our GMCNN contrarily do not have this problem. Our method also overcomes the limitation of the coarse-to-fine architecture, which paints the missing pixels from small to larger scales where errors in the coarse-level already influence refinement. Our GMCNN incorporates different structures in parallel. They complement each other instead of simply inheriting information.

3.2 ID-MRF Regularization

Here, we address aforementioned semantic structure matching and computational-heavy iterative MRF optimization issues. Our scheme is to take MRF-like regularization only in the training phase, named *implicit diversified Markov random fields* (ID-MRF). The proposed network is optimized to minimize the difference between generated content and corresponding nearest-neighbors from ground truth in the feature space. Since we only use it in training, complete ground truth images make it possible to know high-quality nearest neighbors and give appropriate constraints for the network.

To calculate ID-MRF loss, it is possible to simply use direct similarity measure (*e.g.* cosine similarity) to find the nearest neighbors for patches in generated content. But this procedure tends to yield smooth structure, as a flat region easily connects to similar patterns and quickly reduces structure



(a) (b) (c)

Figure 3: Using different similarity measures to search the nearest neighbors. (a) Inpainting results using cosine similarity. (b) Inpainting results using our relative similarity. (c) Ground truth image where red rectangle highlights the filling region (Best viewed in original resolution and with color).

variety, as shown in Figure 3(a). We instead adopt a relative distance measure [17, 16, 22] to model the relation between local features and target feature set. It can restore subtle details as illustrated in Figure 3(b).

Specifically, let $\hat{\mathbf{Y}}_g$ be the generated content for the missing regions, $\hat{\mathbf{Y}}_g^L$ and $\mathbf{Y}^L$ are the features generated by the L_{th} feature layer of a pretrained deep model. For neural patches $\mathbf{v}$ and $\mathbf{s}$ extracted from $\hat{\mathbf{Y}}_g^L$ and $\mathbf{Y}^L$ respectively, the relative similarity from $\mathbf{v}$ to $\mathbf{s}$ is defined as

$$\mathrm{RS}(\mathbf{v},\mathbf{s}) = \exp((\frac{\mu(\mathbf{v},\mathbf{s})}{\max_{\mathbf{r}\in\rho_{\mathbf{s}}(\mathbf{Y}^L)}\mu(\mathbf{v},\mathbf{r})+\epsilon})/h), \tag{1}$$

where $\mu(\cdot,\cdot)$ is the cosine similarity. $\mathbf{r}\in\rho_{\mathbf{s}}(\mathbf{Y}^L)$ means $\mathbf{r}$ belongs to $\mathbf{Y}^L$ excluding $\mathbf{s}$. h and ϵ are two positive constants. If $\mathbf{v}$ is like $\mathbf{s}$ more than other neural patches in $\mathbf{Y}^L$, $\mathrm{RS}(\mathbf{v},\mathbf{s})$ turns large.

Next, $\mathrm{RS}(\mathbf{v},\mathbf{s})$ is normalized as

$$\overline{\mathrm{RS}}(\mathbf{v},\mathbf{s}) = \mathrm{RS}(\mathbf{v},\mathbf{s}) / \sum_{\mathbf{r}\in\rho_{\mathbf{s}}(\mathbf{Y}^L)} \mathrm{RS}(\mathbf{v},\mathbf{r}). \tag{2}$$

Finally, with Eq. (2), the ID-MRF loss between $\hat{\mathbf{Y}}_g^L$ and $\mathbf{Y}^L$ is defined as

$$\mathcal{L}_M(L) = -\log(\frac{1}{Z}\sum_{\mathbf{s}\in\mathbf{Y}^L}\max_{\mathbf{v}\in\hat{\mathbf{Y}}_g^L}\overline{\mathrm{RS}}(\mathbf{v},\mathbf{s})), \tag{3}$$

where Z is a normalization factor. For each $\mathbf{s}\in\mathbf{Y}^L$, $\hat{\mathbf{v}} = \arg\max_{\mathbf{v}\in\hat{\mathbf{Y}}_g^L}\overline{\mathrm{RS}}(\mathbf{v},\mathbf{s})$ means $\hat{\mathbf{v}}$ is closer to $\mathbf{s}$ compared with other neural patches in $\hat{\mathbf{Y}}_g^L$. In the extreme case that all neural patches in $\hat{\mathbf{Y}}_g^L$ are close to one patch $\mathbf{s}$, other patches $\mathbf{r}$ have their $\max_{\mathbf{v}}\overline{\mathrm{RS}}(\mathbf{v},\mathbf{r})$ small. So $\mathcal{L}_M(L)$ is large.

On the other hand, when the patches in $\hat{\mathbf{Y}}_g^L$ are close to different candidates in $\mathbf{Y}^L$, each $\mathbf{r}$ in $\mathbf{Y}^L$ has its unique nearest neighbor in $\hat{\mathbf{Y}}_g^L$. The resulting $\max_{\mathbf{v}\in\hat{\mathbf{Y}}_g^L}\overline{\mathrm{RS}}(\mathbf{v},\mathbf{r})$ is thus big and $\mathcal{L}_M(L)$ becomes small. We show one example in the supplementary file. From this perspective, minimizing $\mathcal{L}_M(L)$ encourages each $\mathbf{v}$ in $\hat{\mathbf{Y}}_g^L$ to approach different neural patches in $\mathbf{Y}^L$, diversifying neighbors, as shown in Figure 3(b).

An obvious benefit for this measure is to improve the similarity between feature distributions in $\hat{\mathbf{Y}}_g^L$ and $\mathbf{Y}^L$. By minimizing the ID-MRF loss, not only local neural patches in $\hat{\mathbf{Y}}_g^L$ find corresponding candidates from $\mathbf{Y}^L$, but also the feature distributions come near, helping capture variation in complicated texture.

Our final ID-MRF loss is computed on several feature layers from VGG19. Following common practice [5, 14], we use *conv4_2* to describe image semantic structures. Then *conv3_2* and *conv4_2*

are utilized to describe image texture as

$$\mathcal{L}_{mrf} = \mathcal{L}_M(conv4\_2) + \sum_{\mathbf{t}=3}^{4} \mathcal{L}_M(conv\mathbf{t}\_2). \tag{4}$$

More Analysis During training, ID-MRF regularizes the generated content based on the reference. It has the strong ability to create realistic texture locally and globally. We note the fundamental difference from the methods of [24, 26], where nearest-neighbor search via networks is employed in the testing phase. Our ID-MRF regularization exploits both reference and contextual information inside and out of the filling regions, and thus causes high diversity in inpainting structure generation.

3.3 Information Fusion

Spatial Variant Reconstruction Loss Pixel-wise reconstruction loss is important for inpainting [18, 25, 26]. To exert constraints based on spatial location, we design the confidence-driven reconstruction loss where unknown pixels close to the filling boundary are more strongly constrained than those away from it. We set the confidence of known pixels as 1 and unknown ones related to the distance to the boundary. To propagate the confidence of known pixels to unknown ones, we use a Gaussian filter g to convolve $\overline{\mathbf{M}}$ to create a loss weight mask $\mathbf{M}_w$ as

$$\mathbf{M}_w^i = (g * \overline{\mathbf{M}}^i) \odot \mathbf{M}, \tag{5}$$

where g is with size 64×64 and its standard deviation is 40. $\overline{\mathbf{M}}^i = \mathbf{1} - \mathbf{M} + \mathbf{M}_w^{i-1}$ and $\mathbf{M}_w^0 = \mathbf{0}$. $\odot$ is the Hadamard product operator. Eq. (5) is repeated several times to generate $\mathbf{M}_w$. The final reconstruction loss is

$$\mathcal{L}_c = ||(\mathbf{Y} - G([\mathbf{X}, \mathbf{M}]; \theta)) \odot \mathbf{M}_w||_1, \tag{6}$$

where $G([\mathbf{X}, \mathbf{M}]; \theta)$ is the output of our generative model G, and θ denotes learn-able parameters.

Compared with the reconstruction loss used in [18, 25, 26], ours exploits spatial locations and their relative order by considering confidence on both known and unknown pixels. It results in the effect of gradually shifting learning focus from filling border to the center and smoothing the learning curve.

Adversarial Loss Adversarial loss is a catalyst in filling missing regions and becomes common in many creation tasks. Similar to those of [9, 26], we apply the improved Wasserstein GAN [6] and use local and global discriminators. For the generator, the adversarial loss is defined as

$$\mathcal{L}_{adv} = -E_{\mathbf{X} \sim \mathbb{P}_{\mathbf{X}}}[D(G(\mathbf{X}; \theta))] + \lambda_{gp} E_{\hat{\mathbf{X}} \sim \mathbb{P}_{\hat{\mathbf{X}}}}[(||\nabla_{\hat{\mathbf{X}}} D(\hat{\mathbf{X}}) \odot \mathbf{M}_w||_2 - 1)^2], \tag{7}$$

where $\hat{\mathbf{X}} = tG([\mathbf{X}, \mathbf{M}]; \theta) + (1 - t)\mathbf{Y}$ and $t \in [0, 1]$.

3.4 Final Objective

With confidence-driven reconstruction loss, ID-MRF loss, and adversarial loss, the model objective of our net is defined as

$$\mathcal{L} = \mathcal{L}_c + \lambda_{mrf}\mathcal{L}_{mrf} + \lambda_{adv}\mathcal{L}_{adv}, \tag{8}$$

where λ_{adv} and λ_{mrf} are used to balance the effects between local structure regularization and adversarial training.

3.5 Training

We train our model first with only confidence-driven reconstruction loss and set λ_{mrf} and λ_{adv} to 0s to stabilize the later adversarial training. After our model G converges, we set $\lambda_{mrf} = 0.05$ and $\lambda_{adv} = 0.001$ for fine tuning until converge. The training procedure is optimized using Adam solver [13] with learning rate $1e-4$. We set $\beta_1 = 0.5$ and $\beta_2 = 0.9$. The batch size is 16.

For an input image $\mathbf{Y}$, a binary image mask $\mathbf{M}$ (with value 0 for known and 1 for unknown pixels) is sampled at a random location. The input image $\mathbf{X}$ is produced as $\mathbf{X} = \mathbf{Y} \odot (\mathbf{1} - \mathbf{M})$. Our model G takes the concatenation of $\mathbf{X}$ and $\mathbf{M}$ as input. The final prediction is $\hat{\mathbf{Y}} = \mathbf{Y} \odot (\mathbf{1} - \mathbf{M}) + G([\mathbf{X}, \mathbf{M}]) \odot \mathbf{M}$. All input and output are linearly scaled within range $[-1, 1]$.

Table 1: Quantitative results on the testing datasets.

Method	Pairs street view-100		ImageNet-200		Places2-2K		CelebA-HQ-2K	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CE [18]	23.49	**0.8732**	**23.56**	**0.9105**	–	–	–	–
MSNPS [24]	24.44	0.8477	20.62	0.7217	–	–	–	–
CA [26]	23.78	0.8588	22.44	0.8917	20.03	0.8539	23.98	0.9441
Ours	**24.65**	0.8650	22.43	0.8939	**20.16**	**0.8617**	**25.70**	**0.9546**

4 Experiments

We evaluate our method on five datasets of Paris street view [18], Places2 [28], ImageNet [19], CelebA [15], and CelebA-HQ [12].

4.1 Experimental Settings

We train our models on the training set and evaluate our model on the testing set (for Paris street view) or validation set (for Places2, ImageNet, CelebA, and CelebA-HQ). In training, we use images of resolution 256×256 with the largest hole size 128×128 in random positions. For Paris street view, places2, and ImageNet, 256×256 images are randomly cropped and scaled from the full-resolution images. For CelebA and CelebA-HQ face datasets, images are scaled to 256×256. All results given in this paper are not post-processed.

Our implementation is with Tensorflow v1.4.1, CUDNN v6.0, and CUDA v8.0. The hardware is with an Intel CPU E5 (2.60GHz) and TITAN X GPU. Our model costs 49.37ms and 146.11ms per image on GPU for testing images with size 256×256 and 512×512, respectively. Using ID-MRF in training phrase costs 784ms more per batch (with 16 images of $256 \times 256 \times 3$ pixels). The total number of parameters of our generator network is 12.562M.

4.2 Qualitative Evaluation

As shown in Figures 8 and 10, compared with other methods, ours gives obvious visual improvement on plausible image structures and crisp textures. The more reasonably generated structures mainly stem from the multi-column architecture and confidence-driven reconstruction loss. The realistic textures are created via ID-MRF regularization and adversarial training by leveraging the contextual and corresponding textures.

In Figures 9, we show partial results of our method and CA [26] on CelebA and CelebA-HQ face datasets. Since we do not apply MRF in a non-parametric manner, visual artifacts are much reduced. It is notable that finding suitable patches for these faces is challenging. Our ID-MRF regularization remedies the problem. Even the face shadow and reflectance can be generated as shown in Figure 9.

Also, our model is trained with arbitrary-location and -size square masks. It is thus general to be applied to different-shape completion as shown in Figures 10 and 1. More inpainting results are in our project website.

4.3 Quantitative Evaluation

Although the generation task is not suitable to be evaluated by peak signal-to-noise ratio (PSNR) or structural similarity (SSIM), for completeness, we still give them on the testing or validation sets of four used datasets for reference. In ImageNet, only 200 images are randomly chosen for evaluation since MSNPS [24] takes minutes to complete a 256×256 size image. As shown in Table 1, our method produces decent results with comparable or better PSNR and SSIM.

We also conduct user studies as shown in Table 2. The protocol is based on large batches of blind randomized A/B tests deployed on the Google Forms platform. Each survey involves a batch of 40 pairwise comparisons. Each pair contains two images completed from the same corrupted input by two different methods. There are 40 participants invited for user study. The participants are asked to select the more realistic image in each pair. The images are all shown at the same resolution (256×256). The comparisons are randomized across conditions and the left-right order is randomized.

Table 2: Result of user study. Each entry is the percentage of cases where results by our approach are judged more realistic than another solution.

	Paris street view	ImageNet	Places2	CelebA	CelebA-HQ
GMCNN > CE [18]	98.1%	88.3%	-	-	-
GMCNN > MSNPS [24]	94.4%	86.5%	-	-	-
GMCNN > CA [26]	84.2%	78.5%	69.6%	99.0%	93.8%

All images are shown for unlimited time and the participant is free to spend as much time as desired on each pair. In all conditions, our method outperforms the baselines.

4.4 Ablation Study

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

Figure 4: Visual comparison of CNNs with different structures. (a) Input image. (b) Single encoder-decoder. (c) Coarse-to-fine structure [26]. (d) GMCNN with the fixed receptive field in all branches. (e) GMCNN with varied receptive fields.

Single Encoder-Decoder *vs.* Coarse-to-Fine *vs.* GMCNN We evaluate our multi-column architecture by comparing with single encode-decoder and coarse-to-fine networks with two sequential encoder-decoder (same as that in [26] except no contextual layer). The single encoder-decoder is just the same as our branch three (**B3**). To minimize the influence of model capacity, we triple the filter sizes in the single encoder-decoder architecture to make its parameter size as close to ours as possible. The loss for these three structures is the same, including confidence-driven reconstruction loss, ID-MRF loss, and WGAN-GP adversarial loss. The corresponding hyper-parameters are the same. The testing results are shown in Figure 4. Our GMCNN structure with varied receptive fields in each branch predicts reasonable image structure and texture compared with single encoder-decoder and coarse-to-fine structure. Additional quantitative experiment is given in Table 3, showing the proposed structure is beneficial to restore image fidelity.

Table 3: Quantitative results of different structures on Paris street view dataset (ED: Encoder-decoder, -f/-v: fixed/varied receptive fields).

Model	ED	Coarse-to-fine	GMCNN-f	GMCNN-v w/o ID-MRF	GMCNN-v
PSNR	23.75	23.63	24.36	24.62	**24.65**
SSIM	0.8580	0.8597	0.8644	**0.8657**	0.8650

Varied Receptive Fields *vs.* Fixed Receptive Field We then validate the necessity of using varied receptive fields in branches. The GMCNN with the same receptive field in each branch turns to using 3 identical third Branches in Figure 2 with filter size 5×5. Figure 4 shows within the GMCNN structure, branches with varied receptive fields give visual more appealing results.

Spatial Discounted Reconstruction Loss *vs.* Confidence-Driven Reconstruction Loss We compare our confidence-driven reconstruction loss with alternative spatial discounted reconstruction loss [26]. We use a single-column CNN trained only with the losses on the Paris street view dataset. The testing results are given in Figure 5. Our confidence-driven reconstruction loss works better.

With and without ID-MRF Regularization We train a complete GMCNN on the Paris street view dataset with all losses and one model that does not involve ID-MRF. As shown in Figure 6, ID-MRF can significantly enhance local details. Also, the qualitative and quantitative changes are given in Table 4 and Figure 7 about how λ_{mrf} affects inpainting performance. Empirically, $\lambda_{mrf} = 0.02 \sim 0.05$ strikes a good balance.

(a) (b) (c) (a) (b) (c)

Figure 5: Visual comparisons of different reconstruction losses. (a) Input image. (b) Spatial discounted loss [26]. (c) Confidence-driven reconstruction loss.

(a) (b) (c) (a) (b) (c)

Figure 6: Visual comparison of results using ID-MRF and not with it. (a) Input image. (b) Results using ID-MRF. (c) Results without using ID-MRF.

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

Figure 7: Visual comparison of results using ID-MRF with different λ_{mrf} (a) Input image. (b) $\lambda_{mrf} = 2$. (c) $\lambda_{mrf} = 0.2$. (d) $\lambda_{mrf} = 0.02$. (e) $\lambda_{mrf} = 0.002$.

Table 4: Quantitative results about how ID-MRF regularizes the inpainting performance.

λ_{mrf}	2	0.2	0.02	0.002
PSNR	24.62	24.53	**24.64**	24.36
SSIM	**0.8659**	0.8652	0.8654	0.8640

5 Conclusion

We have primarily addressed the important problems of representing visual context and using it to generate and constrain unknown regions in inpainting. We have proposed a generative multi-column neural network for this task and showed its ability to model different image components and extract multi-level features. Additionally, the ID-MRF regularization is very helpful to model realistic texture with a new similarity measure. Our confidence-driven reconstruction loss also considers spatially variant constraints. Our future work will be to explore other constraints with location and content.

Limitations Similar to other generative neural networks [18, 24, 26, 25] for inpainting, our method still has difficulties dealing with large-scale datasets with thousands of diverse object and scene categories, such as ImageNet. When data falls into a few categories, our method works best, since the ambiguity removal in terms of structure and texture can be achieved in these cases.

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

Figure 8: Visual comparisons on Paris street view (up) and ImageNet (down). (a) Input image. (b) CE [18]. (c) MSNPS [24]. (d) CA [26]. (e) Our results (best viewed in higher resolution).

(a) (b) (c) (a) (b) (c)

Figure 9: Visual comparisons on CelebA (Left) and CelebA-HQ (Right). (a) Input image. (b) CA [26]. (c) Our results.

(a) (b) (c)

Figure 10: Visual comparisons on Places2 for 512×680 images with random masks. (a) Input image. (b) Results by CA [26]. (c) Our results.

References

[1] F. Agostinelli, M. R. Anderson, and H. Lee. Adaptive multi-column deep neural networks with application to robust image denoising. In *NIPS*, pages 1493–1501, 2013.

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *TOG*, 28(3):24, 2009.

[3] D. Ciregan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*, pages 3642–3649. IEEE, 2012.

[4] A. Criminisi, P. Pérez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *TIP*, 13(9):1200–1212, 2004.

[5] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *CVPR*, pages 2414–2423. IEEE, 2016.

[6] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *NIPS*, pages 5769–5779, 2017.

[7] K. He and J. Sun. Statistics of patch offsets for image completion. In *ECCV*, pages 16–29. Springer, 2012.

[8] K. He and J. Sun. Image completion approaches using the statistics of similar patches. *TPAMI*, 36(12):2423–2435, 2014.

[9] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *TOG*, 36(4):107, 2017.

[10] J. Jia and C.-K. Tang. Image repairing: Robust image synthesis by adaptive nd tensor voting. In *CVPR*, volume 1, pages I–I. IEEE, 2003.

[11] J. Jia and C.-K. Tang. Inference of segmented color and texture description by tensor voting. *TPAMI*, 26(6):771–786, 2004.

[12] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[13] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[14] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *CVPR*, pages 2479–2486, 2016.

[15] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *ICCV*, pages 3730–3738, 2015.

[16] R. Mechrez, I. Talmi, F. Shama, and L. Zelnik-Manor. Learning to maintain natural image statistics. *arXiv preprint arXiv:1803.04626*, 2018.

[17] R. Mechrez, I. Talmi, and L. Zelnik-Manor. The contextual loss for image transformation with non-aligned data. *arXiv preprint arXiv:1803.02077*, 2018.

[18] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, pages 2536–2544, 2016.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum. Image completion with structure propagation. In *TOG*, volume 24, pages 861–868. ACM, 2005.

[22] I. Talmi, R. Mechrez, and L. Zelnik-Manor. Template matching with deformable diversity similarity. In *CVPR*, pages 175–183, 2017.

[23] Z. Yan, X. Li, M. Li, W. Zuo, and S. Shan. Shift-net: Image inpainting via deep feature rearrangement. *arXiv preprint arXiv:1801.09392*, 2018.

[24] 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*, volume 1, page 3, 2017.

[25] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *CVPR*, pages 5485–5493, 2017.

[26] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. *arXiv preprint arXiv:1801.07892*, 2018.

[27] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Single-image crowd counting via multi-column convolutional neural network. In *CVPR*, pages 589–597, 2016.

[28] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *TPAMI*, 2017.