SCAN: Spatial and Channel Attention Normalization For Image Inpainting

Shiyu Chen[1], Wenxin Yu[1*], Liang Nie[1], Xuewen Zhang[1], Siyuan Li[1], Zhiqiang Zhang[1], and Jun Gong[2]

[1] Southwest University of Science and Technology
[2] Beijing Institute of Technology
*yuwenxin@swust.edu.cn

Abstract. Image inpainting focuses on predicting contents with shape structure and consistent details in damaged regions. Recent approaches based on convolutional neural network (CNN) have shown promising results via adversarial learning, attention mechanism, and various loss functions. This paper introduces a novel module named Spatial and Channel Attention Normalization (SCAN), combining attention mechanisms in spatial and channel dimension and normalization to handle complex information of known regions while avoiding its misuse. Compared with the existing self-attention mechanism, our SCAN module produces the attention map adaptively via convolution calculation and then divides the feature map in spatial or channel dimensions to normalize them respectively. Experiments on the CelebA-HQ and Paris Street View datasets indicate that the performance of the proposed method outperforms the current state-of-the-art (SOTA) inpainting approaches. In general, the PSNR of our method is higher 0.65-1.40 dB than SOTA, and the SSIM higher 0.009-0.031.

Keywords: Image Inpainting · Deep Learning · Attention Mechanism · Normalization.

1 Introduction

Image inpainting task as a research hotspot in computer vision has many applications in photo editing, object removal, and image-based rendering. However, because the damaged regions can only be inferred through the known, it is still challenging to synthesize visually realistic and semantically plausible pixels for the damaged regions, coherent with existing ones. Given the circumstances, CNN and generative adversarial networks (GAN) have been introduced to tackle the image inpainting task. The attention mechanism is becoming increasingly popular as a plug-and-play module to encode where to emphasize or suppress. The attention mechanism in spatial dimension explicitly constructs the long-range dependency between the pixels or regions inside and outside the hole via computing their similarity to tackle the ineffectiveness of basic convolutional neural networks in learning long-range information. Also, some approaches model the attention map in channel dimension to enhance those critical features.Feature normalization (FN) is an important technique to help neural network training, typically normalizing features across spatial-dimension, even though it can lead to the shift of mean and variance and mislead training due to the impact of information in damaged regions. Region normalization[20] and attentive normalization[14] still do not solve this issue well, although they both try to divide the feature map into different regions for normalization. A More detailed literature review of the existing image inpainting methods can refer to in the next section.

Motivated by these, we proposed a two-stage network with Spatial and Channel Attention Normalization (SCAN) module to handle these problems. Firstly, in this paper, we choose a two-stage network as our baseline, but different from other two-stage networks[11,18,19] using a series-coupled architecture, the generators of our network use their respective encoders to encode different information and share the same decoder to achieve the same aim—a promising predicted image.

The SCAN module we proposed comprises Spatial Attention Normalization (SAN) and Channel Attention Normalization (CAN). The idea of Spatial Attention Normalization, same as region normalization (RN)[20] and attentive normalization (AN)[14], is to divide the feature map into different regions and then separately normalize them through the learned attention map, improving instance normalization. The idea of Channel Attention Normalization is to group channels of the feature map in bottleneck and then normalize the groups separately, improving group normalization (GN)[16].

The models we presented are evaluated on the test dataset of CelebA-HQ[7] and Paris Street View[1]. Compared with those state-of-the-art inpainting approaches, the produced results quantitatively achieve significant improvement. Our main contributions are as follows:

- We propose a Spatial Attention Normalization method for image inpainting, which will not be disturbed by damaged information or other semantics when normalization.
- We propose a Channel Attention Normalization method that can strengthen the connection between similar semantics and enable semantic separation.
- Experiments on the CelebA-HQ and Paris Street View datasets demonstrate the superiority of our approach compares to the existing advanced approaches.

2 Related Work

2.1 Learning-based Image Inpainting

The methods based on convolutional neural networks[12,5,11] are introduced to help understand the semantic of images during inpainting in the last few years. Pathak et al.[12] first introduce Context Encoder, which uses an encoder-decoder architecture with adversarial training to analyse the high-level semantic information. Iizuka et al.[5] propose global and local discriminators to train the model to get more consistent results in local and global regions. Nazeri et al.[11] divide the training process into two parts, taking the outputs of the first stage as prior structure knowledge to guide image inpainting in the second stage.

However, these approaches let convolution kernels deal with the information inside and outside the hole regions in the same way, which will mislead the encoder. Liu et al.[9] take this issue via exploiting the partial convolutional layer and mask-update operation. After then, Yu et al.[19] present the Gated Convolutional that learns a dynamic mask-update mechanism to replace the hard mask-update and combines it with SN-PatchGAN discriminator to predict better.

There are also some approaches, such as [8], to solve this problem by progressive inpainting the damaged regions from boundary to the centre, improving structure consistency. However, these methods are less practical due to their computational cost.

2.2 Attention Mechanism

The attention modules construct the attention map exploiting the similarity between the pixels of known regions and damaged regions explicitly. Yu et al.[18] propose a contextual attention layer to catch and borrow information of the related patches explicitly at distant spatial locations from the hole. Attention propagation is also incorporated for further coherency of attention. Wang et al.[13] propose a multi-scale attention module to capture information in multiple scales via using attention module of different matching patch sizes. At the same time, Liu et al.[10] introduce a coherent semantic attention layer equivalent to improving the attention propagation to construct the correlation between the deep features of hole regions.

However, these approaches explicitly calculating pair-wise relationship in the feature map demands quadratic complexity (regarding both time and space), limiting its application to large feature maps.

Meanwhile, the interdependencies between channels of feature map in deep layers are also important for the model to understand the semantics of the image. Hu et al.[3] introduce a Squeeze-and-Excitation module to calculate the relationship between channels by exploiting the averages of each channel explicitly. Woo et al.[15] aggregate channel information of a feature map by using max and average pooling operations and then forward them to a shared MLP to produce the attention map. Fu et al.[2] exploit spatial information at all corresponding positions to model channel correlations.

2.3 Normalization

In the inpainting task, Tao et al.[20] exploit Region Normalization to simply separates the feature map into only two regions, uncorrupted region, and corrupted region, according to region mask and normalize them separately to avoid the effect of error information in holes. Yi et al.[14] further propose Attentive

Normalization (AN) to divide the feature map into different semantic regions by the learning semantic map and normalize them separately. AN cannot avoid the possibility of being misled by error information in damaged regions and invalid information in known regions because AN randomly selects n feature pixels from translated feature map as initial semantic filters to guide the learning of the semantic layout map.

3 Method

The Spatial and Channel Attention Normalization module (SCAN) consists of two sub-modules: spatial attention normalization (SAN) and channel attention normalization (CAN). Thus we introduce them respectively in this section. Then we show the overall network architecture and the loss functions.

3.1 Spatial Attention Normalization (SAN)

Due to the reason we mentioned in section 2.3, when normalizing the feature map RN and AN both fail to calculate the mean and variance of each region without shift. The SAN can be divided into three steps, attention map learning, self-sampling regularization, and normalization, as shown in Figure 1. Unlike AN, which is possible to sample the invalid information due to uniform sampling in the global region, our self-sampling regularization distinguishes helpful semantic information. Thus SAN can effectively transfer them from the known regions to damaged regions but avoid being misled by invalid information.

Attention Map Learning The learning of attention map is based on two assumptions. The first one is attention map can indicate where and what to attend in a feature map. The second one is the feature map is composed of n semantic entities, and each pixel from the feature map belongs to at least one of them. For the given feature map $\mathbf{X} \in \mathbb{R}^{c \times h \times w}$ from decoder, we have filters $K_s \in \mathbb{R}^{n \times c \times 1 \times 1}$ to fit the semantic entities through back-propagation, where h, w is the height and width of the feature map, c is the number of channels, and n denotes a predefined number of semantics entities.

We define the correlation between the feature map and these semantic entities as the raw attention map about these semantic entities $\mathbf{S}^{raw}$, of which the calculation could be implemented as a convolution calculation

$$\mathbf{S}^{raw} = K_s(\mathbf{X}) \tag{1}$$

Each channel of the attention map $\mathbf{S}^{raw}$ represents a semantic entity's attention, and the weight of each pixel indicates where to pay attention to. We initially aggregate the feature points from the input feature map into different regions based on the attention map about these entities.

Further, to ensure that these filters learns diverse semantic entities, orthogonal regularization is employed to these entities as

$$\mathcal{L}_{so} = \lambda_{so} \| K_s K_s^T - \mathrm{I} \|_{\mathrm{F}}^2 \tag{2}$$

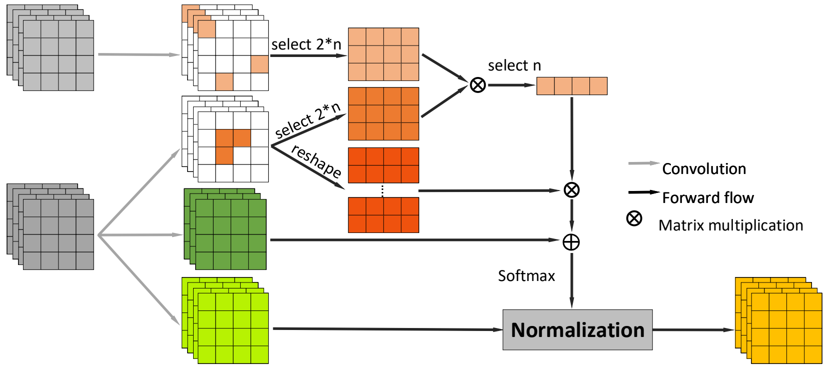

Fig. 1. The proposed SAN module. The selected patches' size is 3×3.

where $K_s \in \mathbb{R}^{n\times c}$ is a weight matrix squeezed to 2-dimension (each row represents the spanned weight of a semantic entity).

Self-sampling Regularization We firstly randomly (uniform sampling) select 2 * n patches from damaged regions of feature map $\mathbf{X}$ and known regions of $\mathbf{F}$, respectively, where $\mathbf{F} \in \mathbb{R}^{c\times h\times w}$ is the feature maps from the encoder located symmetrically to $\mathbf{X}$. We use partial convolution[9] to update the mask to distinguish the damaged regions from known regions and denote the selected patches from them as $s(\mathbf{X})^{2n}$ and $s(\mathbf{F})^{2n}$, respectively. The patch size is chosen as 3×3 to catch a much larger scale of information. Then the correlation between them could be calculated as

$$\varphi_{i,j} =< \frac{s(\mathbf{X})_i^{2n}}{\|s(\mathbf{X})_i^{2n}\|}, \frac{s(\mathbf{F})_j^{2n}}{\|s(\mathbf{F})_j^{2n}\|} > \qquad (3)$$

where i and j denote index of patches.

Next, we select the n patches in $s(\mathbf{F})^{2n}$ with the highest similarity to the damaged regions as regularizing semantic filters $s(\mathbf{F})^n$, according to the relation-map φ.

Finally, the calculation of regularization term $\mathbf{S}^{re}$ could be implemented using $s(\mathbf{F})^n$ as kernels to perform convolution calculations on $\mathbf{X}$.

Normalization With the learned attention map $\mathbf{S}^{raw}$ and attention regularization term $\mathbf{S}^{re}$, the semantics attention map $\mathbf{S}$ are computed as

$$\mathbf{S} = \mathbf{S}^{raw} + \alpha \mathbf{S}^{re} \qquad (4)$$

where $\alpha \in \mathbb{R}^{1\times 1\times n}$ is a learnable vector initialized as 0.1. It adjusts the effects of $\mathbf{S}^{re}$ adaptively, preventing some entities from learning useless semantics.

Then to get attention score map $\mathbf{S}^*$, we apply the softmax operations as

$$\mathbf{S}_j^* = \frac{exp(\mathbf{S}_j)}{\sum_{i=1}^{n} exp(\mathbf{S}_i)} \tag{5}$$

where i and j index the channels. Each $\mathbf{S}_j^*$ is a soft weight map, indicating the probability of every pixel belonging to semantic entity j.

According to the attention score map, we can divide the feature map into n regions and normalization them respectively as

$$\bar{\mathbf{X}} = \sum_{i=1}^{n} \frac{\mathbf{X} - \mu(\mathbf{X}_{\mathbf{S}_i^*})}{\sigma(\mathbf{X}_{\mathbf{S}_i^*}) + \epsilon} \odot \mathbf{S}_i^* \tag{6}$$

where $\mathbf{X}_{\mathbf{S}_i^*} = \mathbf{X} \odot \mathbf{S}_i^*$ and broadcast operations first broadcast $\mathbf{X}$ and $\mathbf{S}^*$ to $\mathbb{R}^{c \times n \times h \times w}$ to match the dimensions of the two matrices. The affine transformation parameter vectors α and β are both dropped. $\mu(\cdot)$ and $\sigma(\cdot)$ compute the mean and standard deviation from instance respectively. Each region shares a mean and variance, strengthening the connection between internal pixels, even if they are far apart.

The final output of SAN compute the weighted sum of the original input feature map and the normalized one as

$$SAN(\mathbf{X}) = \gamma \bar{\mathbf{X}} + \mathbf{X} \tag{7}$$

where γ is a learnable scalar initialized as 0.

3.2 Channel Attention Normalization (CAN)

Each channel of feature map in deep layers can be regarded as a semantic entity response, and different semantic responses (like right eye and left eye) are associated with each other. The commonly used instance normalization (IN)[4] in image inpainting tasks cannot exploit the correlation between channels, meanwhile group normalization(GN)[16] only simply groups the channels. However, CAN groups these channels adaptively through semantic similarity and operate normalization within the group.

Similar to SAN, CAN also can be divided into three steps, attention map learning, self-sampling regularization, and normalization, as shown in Figure 2.

Attention Map Learning For given feature map $\mathbf{X} \in \mathbb{R}^{c \times h \times w}$, we calculate the attention map $\mathbf{C}$ by convolution calculation using $K_c \in \mathbb{R}^{n \times c \times 1 \times 1}$ as filters, similar to Eq. (1). We still have $\mathcal{L}_{co}$ to guide K_c, which is calculated similarly to Eq. (2).

In order to divide the channels of the feature map into various groups according to their semantics, we continue to calculate the correlation between $\mathbf{X}$ and the attention map $\mathbf{C}$ as the raw grouping basis $\mathbf{R}^{raw}$. Specifically, we reshape $\mathbf{C}$ to $\mathbb{R}^{n \times hw}$ and $\mathbf{X}$ to $\mathbb{R}^{c \times hw}$, and then perform a matrix multiplication between $\mathbf{C}$ and the transpose of $\mathbf{X}$.

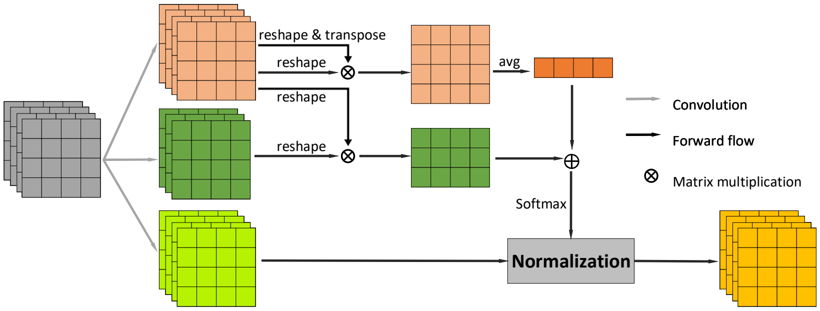

Fig. 2. The proposed CAN module.

Self-attention Sampling Different from SAN, we first calculate the correlation between channels and then calculate its average value as

$$\mathbf{R}_j^{re} = \frac{\sum_{i=0,i\neq j}^{c} \mathbf{X}_i(\mathbf{X}_j)^T}{c} \tag{8}$$

where $\mathbf{X}$ is reshaped to $\mathbb{R}^{c\times hw}$. $\mathbf{R}_j^{re}$ represents the average correlation between the j-th channels and other channels, $j \in [1, c]$. $\mathbf{R}^{re}$ is broadcasted to $\mathbb{R}^{n\times c}$ as the regularization term.

Normalization Finally, we get the regularized grouping basis $\mathbf{R}$ by summing $\mathbf{R}^{raw}$ and $\mathbf{R}^{re}$ which is weighted with a learnable vector $\beta \in \mathbb{R}^{1\times 1\times n}$ initialized to 0.1. Then we apply softmax to obtain the soft grouping basis,

$$x_{i,j}^* = \frac{exp(x_{i,j})}{\sum_{i=1}^{n} exp(x_{i,j})} \tag{9}$$

where $j \in [1, c]$. Each $x_{i,j}^*$ in $\mathbf{R}^* \in \mathbb{R}^{n\times c}$ measures the possibility that the i-th channel of the feature map belongs to the j-th group.

Then we divide the feature map into n groups in channel dimension and normalize them respectively as

$$\bar{\mathbf{X}} = \sum_{i=1}^{n} \frac{\mathbf{X} - \mu(\mathbf{X}_{\mathbf{R}_i^*})}{\sigma(\mathbf{X}_{\mathbf{R}_i^*}) + \epsilon} \odot \mathbf{R}_i^* \tag{10}$$

where $\mathbf{X}_{\mathbf{R}^*} = \mathbf{X} \odot \mathbf{R}^*$, and broadcast operations first broadcast we broadcast $\mathbf{X}$ and $\mathbf{R}^*$ to $\mathbb{R}^{n\times c\times h\times w}$ to match the dimensions of the two matrices.

The final output of CAN compute the weighted sum of the original input feature map and the normalized one as

$$CAN(\mathbf{X}) = \eta\bar{\mathbf{X}} + \mathbf{X} \tag{11}$$

where η is a learnable scalar initialized as 0.

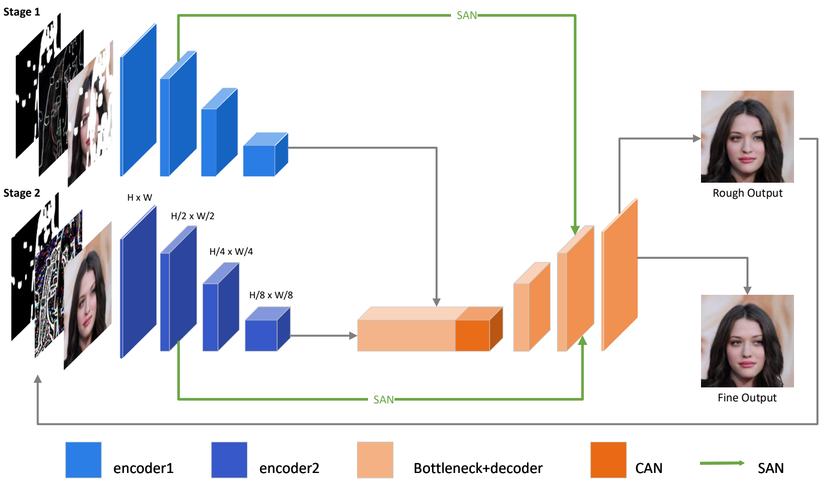

Fig. 3. The overview of our network architecture.

3.3 Networks

The overall network architecture with SCAN is shown in Figure 3. Formally, let $\mathbf{I}$ be the ground truth image, $\mathbf{G}$ and $\mathbf{H}$ denote its gradient and high-frequency residual map.

The task of the first stage is to generate a rough result with good structure information but insufficient texture details. Thus, we take the masked image $\hat{\mathbf{I}} = \mathbf{I} \odot \mathbf{M}$ as the input and corresponding gradient map $\hat{\mathbf{G}} = \mathbf{G} \odot \mathbf{M}$, In addition with the image mask $\mathbf{M}$ (with value 1 for known region otherwise 0) as preconditions. Here, $\odot$ denotes the Hadamard product.

The generator produces a rough result $\mathbf{I}_{rough} = G_1(\hat{\mathbf{I}}, \hat{\mathbf{G}}, \mathbf{M})$ as part of the input of second stage, where G_1 represents the generator of the first stage, which is composed of enc_1 and dec.

The second-stage generator has its own independent encoder but uses a shared decoder with the first stage because the information that needs to be focused on is different. The first stage focuses on processing structural information, and the second stage is texture detail information. Specifically, the masked high-frequency residual map $\hat{\mathbf{H}} = \mathbf{H} \odot \mathbf{M}$, instead of the gradient map, is another part of the input. In addition, smaller convolution kernels are used in the second stage's encoder than in the first stage's.

We get the fine image $\mathbf{I}_{fine} = G_2(\hat{\mathbf{I}} + \mathbf{I}_{rough} \odot (1 - \mathbf{M}), \hat{\mathbf{H}}, \mathbf{M})$, where G_2 represents the generator of the two stage, which is composed of enc_2 and dec. The final predicted image is $\mathbf{I}_{pred} = \hat{\mathbf{I}} + \mathbf{I}_{fine} \odot (1 - \mathbf{M})$.

In addition, we use the attention map calculated by SAN to make attention transfer[17] which enables filling holes by weighted copying features from context. Taking gradient map as an example, we select n patches (denote it as $s(\mathbf{G})^n$) at the same position as $s(\mathbf{F})^n$ and then reshape it to $\mathbb{R}^{c \times n \times p \times p}$, where p is the selected patches' size (in our experiment is 3), and use it as convolution filters to reconstruct the gradient information in damaged regions. Finally, we concatenate it with the decoding feature map.

3.4 Loss Functions

The Generator is trained over a joint loss that consists of an orthogonal loss, ℓ_1 loss, perceptual loss, style loss, and adversarial loss.

The $\mathcal{L}_o$ calculates the sum of $\mathcal{L}_{so}$ and $\mathcal{L}_{co}$ of two-stage, enabling attention map can learn various semantic attention.

Similarly, $\mathcal{L}_{\ell_1}$ is the sum of the distance between $\mathbf{I}_{pred}$, $\mathbf{G}_{pred}$, and $\mathbf{H}_{pred}$ and their corresponding Ground Truth. To ensure proper scaling, the ℓ_1 loss is normalized by the mask size.

We also calculate the perceptual loss[6] and the style loss to enable the model to learn high-level representations and remove artifacts caused by deconvolution. Perceptual loss and style loss are respectively defined as

$$\mathcal{L}_{perc} = \mathbb{E}\left[\sum_i \frac{1}{N_i} \left\|\phi_i\left(\mathbf{I}\right) - \phi_i\left(\mathbf{I}_{pred}\right)\right\|_1\right] \tag{12}$$

$$\mathcal{L}_{style} = \mathbb{E}_j\left[\left\|G_j^\phi\left(\mathbf{I}\right) - G_j^\phi\left(\mathbf{I}_{pred}\right)\right\|_1\right] \tag{13}$$

where ϕ_i is the activation map of the i-th selected layer from VGG-19 and G_j^ϕ is a $c_j \times c_j$ Gram matrix constructed from activation maps ϕ_j (c_j is the number of channels of ϕ_j). Here, layers $relu1\_1$, $relu2\_1$, $relu3\_1$, $relu4\_1$. $relu5\_1$ are used.

We take the PatchGAN as our discriminator D to produce more convincing results and denote the adversarial loss for our generator as

$$\mathcal{L}_G = \mathbb{E}_{(\mathbf{I}_{pred})} log[1 - D(\mathbf{I}_{pred})] \tag{14}$$

In summary, the overall loss function of the proposed SCAN algorithm is as follows:

$$\mathcal{L}_{total} = 10\mathcal{L}_o + \mathcal{L}_{\ell_1} + 0.1\mathcal{L}_{prec} + 250\mathcal{L}_{style} + 0.1\mathcal{L}_G \tag{15}$$

4 Experiments

All of the experiments in this paper are conducted in the dataset of CelebA-HQ[7] and Paris Street View[1]. The CelebA-HQ dataset is a high-quality version of CelebA that consists of 28000 train images and 2000 test images. The Paris Street View contains 14900 training images and 100 test images. We use Sobel filters to extract the gradient map and use the result after the image subtracting its Gaussian blur to get the high-frequency residual map. The irregular mask dataset used in this paper comes from the work of Liu[9]. We train the proposed model and the compared models on a single NVIDIA 2080Ti with a batch size of 8 until the generators converge, using Adam optimizer.

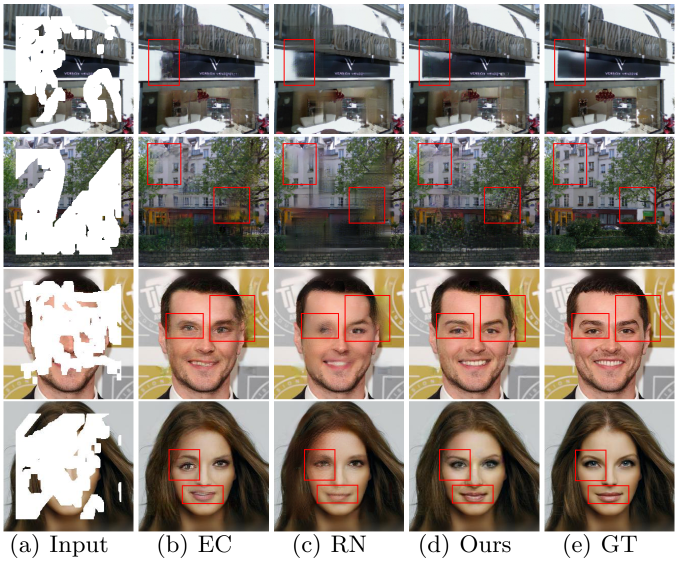

Fig. 4. Qualitative comparisons with EC and RN. The top two rows are the results on Paris Street View, and the bottom two row are the results on CelebA-HQ. The first and third rows are the results when the mask ratio is 30%-40%, and the second row and fourth rows are 40%-50%.

4.1 Quantitative Results

The Quantitative results in the test dataset of CelebA-HQ and Paris Street View are shown in Table 1, where some results produced by popular inpainting methods in comparison are also shown. In the case of different ratios of the damaged region, the table demonstrates the inpainting ability of our network, showing that our results are better than other results in PSNR (Peak Signal-to-Noise) and SSIM (Structural Similarity) metrics.

In particular, Table 2 demonstrates our method has a more remarkable improvement in all metrics in detail when the mask ratio is 30%-40% (because this ratio is the median value of the mask ratio and the most common in the application). Compared with PC, GC, EC, RN, and RFR, our method can effectively use the correlation with the limited known regions information.

4.2 Qualitative Results

Figure 4 illustrates the visual inpainting results of different methods on the test set of CelebA-HQ and Paris Street View with mask ratios of 30%-40% and 40%-50%.

EC is misled by the edge information generated in the first stage when predicting the results. RN tends to produce smoother results and lacks texture details. Compared with EC and RN, the proposed method can better handle

Table 1. The comparison of PSNR and SSIM over the CelebA-HQ and Paris Street View. EC means Edge Connect method[11], and RN represents Region Normalization approach[20]. Both quantitative evaluations are higher is better.

Dataset		CelebA-HQ			Paris Street View		
mask ratio		20%-30%	30%-40%	40%-50%	20%-30%	30%-40%	40%-50%
PSNR	EC	27.18	25.29	23.33	27.28	25.80	24.10
	RN	27.43	25.51	23.66	27.57	26.19	24.26
	Ours	**28.54**	**26.91**	**24.69**	**28.76**	**27.09**	**25.27**
SSIM	EC	0.917	0.902	0.862	0.875	0.851	0.804
	RN	0.929	0.912	0.871	0.887	0.864	0.812
	Ours	**0.953**	**0.933**	**0.894**	**0.918**	**0.873**	**0.840**

Table 2. The comparison of PSNR, SSIM, and MAE (Mean Absolute Error) over the Paris, in case of 30%-40% mask ratio. †Higher is better. △Lower is better.

	PC[9]	GC[19]	EC[11]	RN[20]	RFR[8]	Ours
PSNR †	25.46	25.54	25.80	26.19	26.44	**27.09**
SSIM †	0.835	0.849	0.851	0.864	0.862	**0.873**
MAE(%) △	3.13	3.09	2.90	3.04	2.75	**2.53**

much larger damaged regions and achieves better subjective results, even houses with complex structural information on Paris Street View. Also, the visual comparison on CelebA-HQ shows obvious enhancement of our method, such as sharp facial contours, crisp eyes and ears, and reason-able object boundaries.

The point to note is that the better qualitative results generated by our method benefit from SAN separating the semantics in the spatial dimension, eliminating the impact of normalization shifts, and CAN strengthening the connection between similar semantics in the channel dimension.

5 Conclusions

This paper proposed a two-stage image inpainting approach with a SCAN module. The SAN and CAN module normalize feature map in spatial and channel dimensions, respectively. Various experiments show that the proposed SCAN generates promising images and achieves the state-of-the-art performance. Compared with the existing methods, our method improves the PSNR by 0.65-1.40 dB and the SSIM by 0.009-0.031. The SAN can avoid shifts caused by invalid information when normalization in the inpainting tasks. In addition, the CAN module improving the group normalization could also be generalized to similar image restoration tasks, including image denoising, conditional image generation, and image segmentation.

References

1. Doersch, C., Singh, S., Gupta, A., Sivic, J., Efros, A.: What makes paris look like paris? ACM Transactions on Graphics **31**(4), 101 (2012)

2. Fu, J., Liu, J., Tian, H., Li, Y., Bao, Y., Fang, Z., Lu, H.: Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3146–3154 (2019)
3. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
4. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1501–1510 (2017)
5. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Globally and locally consistent image completion. ACM Transactions on Graphics (ToG) **36**(4), 1–14 (2017)
6. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711 (2016)
7. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
8. Li, J., Wang, N., Zhang, L., Du, B., Tao, D.: Recurrent feature reasoning for image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7760–7768 (2020)
9. Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 85–100 (2018)
10. Liu, H., Jiang, B., Xiao, Y., Yang, C.: Coherent semantic attention for image inpainting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4170–4179 (2019)
11. Nazeri, K., Ng, E., Joseph, T., Qureshi, F.Z., Ebrahimi, M.: Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212 (2019)
12. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2536–2544 (2016)
13. Wang, N., Li, J., Zhang, L., Du, B.: Musical: Multi-scale image contextual attention learning for inpainting. In: IJCAI. pp. 3748–3754 (2019)
14. Wang, Y., Chen, Y.C., Zhang, X., Sun, J., Jia, J.: Attentive normalization for conditional image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5094–5103 (2020)
15. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
16. Wu, Y., He, K.: Group normalization. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018)
17. Yi, Z., Tang, Q., Azizi, S., Jang, D., Xu, Z.: Contextual residual aggregation for ultra high-resolution image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7508–7517 (2020)
18. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5505–5514 (2018)
19. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4471–4480 (2019)
20. Yu, T., Guo, Z., Jin, X., Wu, S., Chen, Z., Li, W., Zhang, Z., Liu, S.: Region normalization for image inpainting. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 12733–12740 (2020)