Region Normalization for Image Inpainting

Tao Yu, Zongyu Guo, Xin Jin, Shilin Wu, Zhibo Chen,[*] **Weiping Li, Zhizheng Zhang, Sen Liu**

CAS Key Laboratory of Technology in Geo-spatial Information Processing and Application System,
University of Science and Technology of China
{yutao666, guozy, jinxustc, shilinwu, zhizheng}@mail.ustc.edu.cn, {chenzhibo, wpli}@ustc.edu.cn,
elsen@iat.ustc.edu.cn

Abstract

Feature Normalization (FN) is an important technique to help neural network training, which typically normalizes features across spatial dimensions. Most previous image inpainting methods apply FN in their networks without considering the impact of the corrupted regions of the input image on normalization, *e.g.* mean and variance shifts. In this work, we show that the mean and variance shifts caused by full-spatial FN limit the image inpainting network training and we propose a spatial region-wise normalization named Region Normalization (RN) to overcome the limitation. RN divides spatial pixels into different regions according to the input mask, and computes the mean and variance in each region for normalization. We develop two kinds of RN for our image inpainting network: (1) Basic RN (RN-B), which normalizes pixels from the corrupted and uncorrupted regions separately based on the original inpainting mask to solve the mean and variance shift problem; (2) Learnable RN (RN-L), which automatically detects potentially corrupted and uncorrupted regions for separate normalization, and performs global affine transformation to enhance their fusion. We apply RN-B in the early layers and RN-L in the latter layers of the network respectively. Experiments show that our method outperforms current state-of-the-art methods quantitatively and qualitatively. We further generalize RN to other inpainting networks and achieve consistent performance improvements.

1 Introduction

Image inpainting aims to reconstruct the corrupted (or missing) regions of the input image. It has many applications in image editing such as object removal, face editing and image disocclusion. A key issue in image inpainting is to generate visually plausible content in the corrupted regions.

Existing image inpainting methods can be divided into two groups: traditional and learning-based methods. The traditional methods fill the corrupted regions by diffusion-based methods (Bertalmio et al. 2000; Ballester et al. 2001; Esedoglu and Shen 2002; Bertalmio et al. 2003) that propagate neighboring information into them, or patch-based methods (Drori, Cohen-Or, and Yeshurun 2003; Barnes et al.
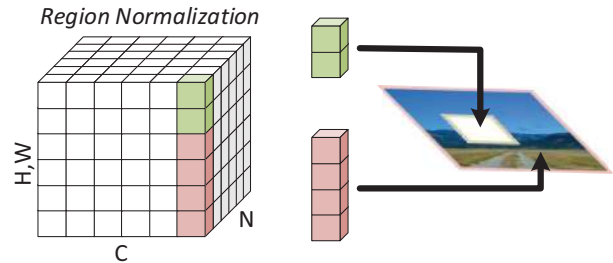
Figure 1: Illustration of our Region Normalization (RN) with region number $K = 2$. Pixels in the same color (green or pink) are normalized by the same mean and variance. The corrupted and uncorrupted regions of the input image are normalized by different means and variances.

2009; Xu and Sun 2010; Darabi et al. 2012) that copy similar patches into them. The learning-based methods commonly train neural networks to synthesize content in the corrupted regions, which yield promising results and have significantly surpassed the traditional methods in recent years. Recent image inpainting works, such as (Yu et al. 2018; Liu et al. 2018; Yu et al. 2019; Nazeri et al. 2019), focus on the learning-based methods. Most of them design an advanced network to improve the performance, but ignore the inherent nature of image inpainting problem: unlike the input image of general vision task, the image inpainting input image has corrupted regions that are typically independent of the uncorrupted regions. Inputing a corrupted image as a general spatially consistent image into a neural network has potential problems, such as convolution of invalid (corrupted) pixels and mean and variance shifts of normalization. Partial convolution (Liu et al. 2018) is proposed to solve the invalid convolution problem by operating on only valid pixels, and achieves a performance boost. However, none of existing methods solve the mean and variance shift problem of normalization in inpainting networks. In particular, most existing methods apply feature normalization (FN) in their networks to help training, and existing FN methods typically normalize features across spatial dimensions, ignoring the corrupted regions and resulting in mean and variance shifts of normalization.

In this work, we show in theory and experiment that the mean and variance shifts caused by existing full-spatial normalization limit the image inpainting network training. To overcome the limitation, we propose Region Normalization (RN), a spatially region-wise normalization method that divides spatial pixels into different regions according to the input mask and computes the mean and variance in each region for normalization. RN can effectively solve the mean and variance shift problem and improve the inpainting network training.

We further design two kinds of RN for our image inpainting network: Basic RN (RN-B) and Learnable RN (RN-L). In the early layers of the network, the input image has large corrupted regions, which results in severe mean and variance shifts. Thus we apply RN-B to solve the problem by normalizing corrupted and uncorrupted regions separately. The input mask of RN-B is obtained from the original inpainting mask. After passing through several convolutional layers, the corrupted regions are fused gradually, making it difficult to obtain a region mask from the original mask. Therefore, we apply RN-L in the latter layers of the network, which learns to detect potentially corrupted regions by utilizing the spatial relationship of the input feature and generates a region mask for RN. Additionally, RN-L can also enhance the fusion of corrupted and uncorrupted regions by global affine transformation. RN-L not only solves the mean and variance shift problem, but also boosts the reconstruction of corrupted regions.

We conduct experiments on Places2 (Zhou et al. 2017) and CelebA (Liu et al. 2015) datasets. The experimental results show that, with the help of RN, a simple backbone can surpass current state-of-the-art image inpainting methods. In addition, we generalize our RN to other inpainting networks and yield consistent performance improvements.

Our contributions in this work include:

- Both theoretically and experimentally, we show that existing full-spatial normalization methods are sub-optimal for image inpainting.
- To the best our knowledge, we are the first to propose spatially region-wise normalization $i.e.$ Region Normalization (RN).
- We propose two kinds of RN for image inpainting and the use of them for achieving state-of-the-art on image inpainting.

2 Related Work

2.1 Image Inpainting

Previous works in image inpainting can be divided into two categories: traditional and learning-based methods.

Traditional methods use diffusion-based (Bertalmio et al. 2000; Ballester et al. 2001; Esedoglu and Shen 2002; Bertalmio et al. 2003) or patch-based (Drori, Cohen-Or, and Yeshurun 2003; Barnes et al. 2009; Xu and Sun 2010; Darabi et al. 2012) methods to fill the holes. The former propagate neighboring information into holes. The latter typically copy similar patches into the holes. The performance of these traditional methods is limited since they cannot use semantic information.

Learning-based methods can learn to extract semantic information by massive data training, and thus significantly improve the inpainting results. These methods map a corrupted image directly to the completed image. ContextEncoder (Pathak et al. 2016), one of pioneer learning-based methods, trains a convolutional neural network to complete image. With the introduction of generative adversarial networks (GANs) (Goodfellow et al. 2014), GAN-based methods (Yeh et al. 2017; Iizuka, Simo-Serra, and Ishikawa 2017; Yu et al. 2018; Xiong et al. 2019; Nazeri et al. 2019) are widely used in image inpainting. ContextualAttention (Yu et al. 2018) is a popular model with coarse-to-fine architecture. Considering that there are valid/uncorrupted and invalid/corrupted regions in a corrupted image, partial convolution (Liu et al. 2018) operates on only valid pixels and achieves promising results. Gated convolution (Yu et al. 2019) generalizes PConv by a soft distinction of valid and invalid regions. EdgeConnect (Nazeri et al. 2019) first predicts the edges of the corrupted regions, then generates the completed image with the help of the predicted edges.

However, most existing inpainting methods ignore the impact of corrupted regions of the input image on normalization which is a crucial technique for network training.

2.2 Normalization

Feature normalization layer has been widely applied in deep neural networks to help network training.

Batch Normalization (BN) (Ioffe and Szegedy 2015), normalizing activations across batch and spatial dimensions, has been widely used in discriminative networks for speeding up convergence and improve model robustness, and found also effective in generative networks. Instance Normalization (IN) (Ulyanov, Vedaldi, and Lempitsky 2016), distinguished from BN by normalizing activations across only spatial dimensions, achieves a significant improvement in many generative tasks such as style transformation. Layer Normalization (LN) (Ba, Kiros, and Hinton 2016) normalizes activations across channel and spatial dimensions ($i.e.$ normalizes all features of an instance), which helps recurrent neural network training. Group Normalization (GN) (Wu and He 2018) normalizes features of grouped channels of an instance and improves the performance of some vision tasks such as object detection.

Different from a single set of affine parameters in the above normalization methods, conditional normalization methods typically use external data to reason multiple sets of affine parameters. Conditional instance normalization (CIN) (Dumoulin, Shlens, and Kudlur 2016), adaptive instance normalization (AdaIN) (Huang and Belongie 2017), conditional batch normalization (CBN) (De Vries et al. 2017) and spatially adaptive denormalization (SPADE) (Park et al. 2019) have been proposed in some image synthesis tasks.
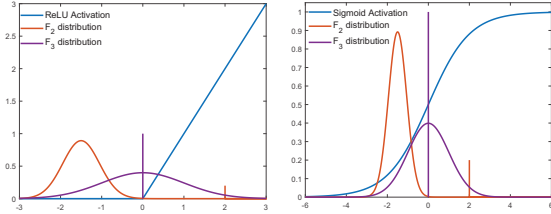
None of existing normalization methods considers spatial distribution's impact on normalization.

3 Approach

In this secetion, we show that existing full-spatial normalization methods are sub-optimal for image inpianting problem as motivation for Region Normalization (RN). We then
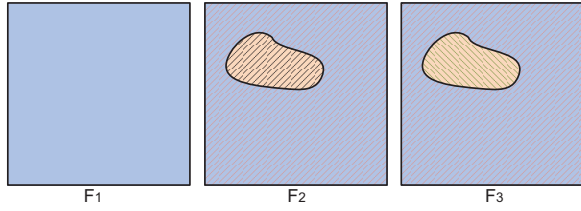
(a) Illustration of three feature maps

(b) Distributions after different normalization

Figure 2: (a) F_1 is the original feature map. F_2 with mask performs full-spatial normalization in all the regions. F_3 performs separate normalization in the masked and unmasked regions. (b) The distribution of F_2's unmasked area has a shift to the nonlinear region, which easily causes the vanishing gradient problem. But F_3 does not have this problem.

introduce two kinds of RN for image inpainting, Basic RN (RN-B) and Learnable RN (RN-L). We finally introduce our image inpainting network using RN.

3.1 Motivation for Region Normalization

Problem in Normalization. F_1, F_2 and F_3 are three feature maps of the same size, each with n pixels, as shown in Figure 2. F_1 is the original uncorrupted feature map. F_2 and F_3 are the different normalization results of feature map with masked and unmasked areas. n_m and n_u are the pixel numbers of the masked and unmasked areas, respectively. Then $n = n_m + n_u$. Specifically, F_2 is normalized in all the areas. F_3 is normalized separately in the masked and unmasked areas. Assuming the masked region pixels have the max value 255, the mean and standard deviation of three feature maps are listed as μ_1, μ_2, μ_{3m}, μ_{3u}, σ_1, σ_2, σ_{3m} and σ_{3u}. The subscripts 1 and 2 represent the entire areas of F_1 and F_2, and $3m$ and $3u$ represent the masked and unmasked areas of F_3, respectively. The relationships are listed below:

$$\mu_{3u} = \mu_1, \sigma_{3u} = \sigma_1 \tag{1}$$

$$\mu_{3m} = 255, \sigma_{3m} = 0 \tag{2}$$

$$\mu_2 = \frac{n_u}{n} * \mu_{3u} + \frac{n_m}{n} * 255 \tag{3}$$

$$\sigma_2{}^2 = \frac{n_u}{n}\sigma_{3u}{}^2 + \frac{n_m * n_u}{n^2}(\mu_{3u} - 255)^2 \tag{4}$$

After normalizing the masked and unmasked areas together, F_2 unmasked area's mean has a shift toward -255 and its variance increases compared with F_1 and F_3. According to (Ioffe and Szegedy 2015), the normalization shifts and scales the distribution of features into a small region where

the mean is zero and the variance is one. We take batch normalization (BN) as an example here. For each point x_i

$$x_i' = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{5}$$

$$y_i = \gamma x_i' + \beta = BN_{\gamma,\beta}(x_i) \tag{6}$$

Compared with the F_3's unmasked area, distribution of F_2's unmasked area narrows down and shifts from 0 toward -255. Then, for both fully-connected and convolutional layer, the affine transformation is followed by an element-wise nonlinearity (Ioffe and Szegedy 2015):

$$z = g(BN(Wu)) \tag{7}$$

Here $g(\cdot)$ is the nonlinear activation function such as ReLU or sigmoid. The BN transform is added immediately before the function, by normalizing $x = Wu + b$. The W and b are learned parameters of the model.

As shown in Figure 2, in the ReLU and sigmoid activations, the distribution region of F_2 is narrowed down and shifted by the masked area, which adds the internal covariate shift and easily get stuck in the saturated regimes of nonlinearities (causing the vanishing gradient problem), wasting lots of time for γ, β and W to fix the problem. However, F_3, normalized the masked and unmasked regions separately, reduces the internal covariate shift, which preserves the network capacity and improves training efficiency.

Motivated by this, we design a spatial region-wise normalization named Region Normalization (RN).

Formulation of Region Normalization. Let $X \in \mathbb{R}^{N \times C \times H \times W}$ be the input feature. N, C, H and W are batch size, number of channels, height and width, respectively. Let $x_{n,c,h,w}$ be a pixel of X and $X_{n,c} \in \mathbb{R}^{H \times W}$ be a channel of X where (n, c, h, w) is an index along (N, C, H, W) axis. Given a region label map (mask) M, $X_{n,c}$ is divided into K regions as follows:

$$X_{n,c} = R_{n,c}^1 \cup R_{n,c}^2 \cup ... \cup R_{n,c}^K \tag{8}$$

The mean and standard deviation of each region of a channel $R_{n,c}^k$ computed by:

$$\mu_{n,c}^k = \frac{1}{|R_{n,c}^k|} \sum_{x_{n,c,h,w} \in R_{n,c}^k} x_{n,c,h,w} \tag{9}$$

$$\sigma_{n,c}^k = \sqrt{\frac{1}{|R_{n,c}^k|} \sum_{x_{n,c,h,w} \in R_{n,c}^k} (x_{n,c,h,w} - \mu_{n,c}^k)^2 + \epsilon} \tag{10}$$

Here k is a region index, $|R_{n,c}^k|$ is the number of pixels in region $R_{n,c}^k$ and ϵ is a small constant. The normalization of each region performs the following computation:

$$\hat{R}_{n,c}^k = \frac{1}{\sigma_{n,c}^k}(R_{n,c}^k - \mu_{n,c}^k) \tag{11}$$

RN merges all normalized regions and obtains the region normalized feature as follows:

$$\hat{X}_{n,c} = \hat{R}_{n,c}^1 \cup \hat{R}_{n,c}^2 \cup ... \cup \hat{R}_{n,c}^K \tag{12}$$

After normalization, each region is transformed separately with a set of learnable affine parameters (γ_c^k, β_c^k).

Analysis of Region Normalization. RN is an alternative to Instance Normalization (IN). RN degenerates into IN when region number K equals to one. RN normalizes spatial regions on each channel separately as the spatial regions are not entirely dependent. We set $K = 2$ for image inpainting in this work, as there are two obviously independent spatial regions in the input image: corrupted and uncorrupted regions. RN with $K = 2$ is illustrated in Figure 1.

3.2 Basic Region Normalization

Basic RN (RN-B) normalizes and transforms corrupted and uncorrupted regions separately. This can solve the mean and variance shift problem of normalization and also avoid information mixing in affine transformation. RN-B is designed for using in early layers of the inpainting network, as the input feature has large corrupted regions, which causes severe mean and variance shifts.

Given an input feature $\boldsymbol{F} \in \mathbb{R}^{C \times H \times W}$ and a binary region mask $\boldsymbol{M} \in \mathbb{R}^{1 \times H \times W}$ indicating corrupted region, RN-B layer first separates each channel $\boldsymbol{F_c} \in \mathbb{R}^{1 \times H \times W}$ of input feature $\boldsymbol{F}$ into two regions R_c^1 (*e.g.* uncorrupted region) and R_c^2 (*e.g.* corrupted region) according to region mask $\boldsymbol{M}$. Let $x_{c,h,w}$ represent a pixel of $\boldsymbol{F_c}$ where (c, h, w) is an index of (C, H, W) axis. The separation rule is as follow:

$$x_{c,h,w} \in \begin{cases} R_c^1 & \text{if } \boldsymbol{M}(h, w) = 1 \\ R_c^2 & \text{otherwise} \end{cases} \tag{13}$$

RN-B then normalizes each region following Formula (9), (10) and (11) with region number $K = 2$. Then we merge the two normalized regions $\hat{R}_c^1$ and $\hat{R}_c^2$ to obtain normalized channel $\hat{\boldsymbol{F}}_{\boldsymbol{c}}$. RN-B is a basic implement of RN and the region mask is obtained from the original inpainting mask.

For each channel, there are two sets of learnable parameters (γ_c^1, β_c^1) and (γ_c^2, β_c^2) for affine transformation of each region. For ease of denotation, we denote $[\gamma_c^1, \gamma_c^2]$ as $\boldsymbol{\gamma}$, $[\beta_c^1, \beta_c^2]$ as $\boldsymbol{\beta}$. RN-B layer is showed in Figure 3(a).

3.3 Learnable Region Normalization

After passing through several convolutional layers, the corrupted regions are fused gradually and obtaining an accurate region mask from the original mask is hard. RN-L addresses the issue by automatically detecting corrupted regions and obtaining a region mask. To further improve the reconstruction, RN-L enhances the fusion of corrupted and uncorrupted regions by global affine transformation. RN-L boosts the corrupted region reconstruction in a soft way, which solves the mean and variance shift problem and also enhances the fusion. Therefore, RN-L is suitable for latter layers of the network. Note that, RN-L does not need a region mask and the affine parameters of RN-L are pixel-wise. RN-L is illustrated in Figure 3(b).

RN-L generates a spatial response map by taking advantage of the spatial relationship of the features themselves. Specifically, RN-L first performs max-pooling and average-pooling along the channel axis. The two pooling operations are able to obtain an efficient feature descriptor (Zagoruyko and Komodakis 2016; Woo et al. 2018). RN-L then concatenates the two pooling results. RN-L is convolved on the two
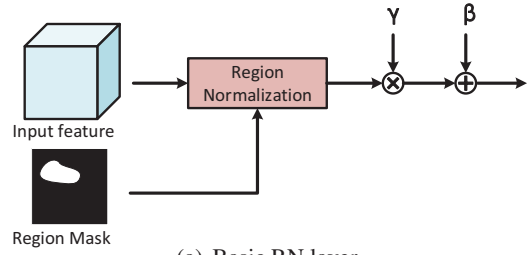
(a) Basic RN layer
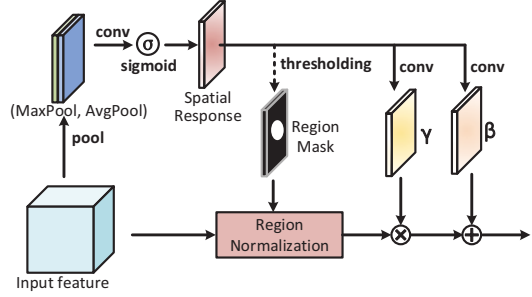
(b) Learnable RN layer

Figure 3: Two kinds of RN: RN-B (a) and RN-L (b)

maps with sigmoid activation to get a spatial response map. The spatial response map is computed as:

$$\boldsymbol{M_{sr}} = \sigma(Conv([\boldsymbol{F_{max}}, \boldsymbol{F_{avg}}])) \tag{14}$$

Here $\boldsymbol{F_{max}} \in \mathbb{R}^{1 \times H \times W}$ and $\boldsymbol{F_{avg}} \in \mathbb{R}^{1 \times H \times W}$ are the max-pooling and average-pooling results of the input feature $\boldsymbol{F} \in \mathbb{R}^{C \times H \times W}$. $Conv$ is the convolution operation and σ is the sigmoid function. $\boldsymbol{M_{sr}} \in \mathbb{R}^{1 \times H \times W}$ is the spatial response map. To get a region mask $\boldsymbol{M} \in \mathbb{R}^{1 \times H \times W}$ for RN, we set a threshold t to the spatial response map:

$$\boldsymbol{M}(h, w) = \begin{cases} 1 & \text{if } \boldsymbol{M_{sr}}(h, w) > t \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

We set threshold $t = 0.8$ in this work. Note that the thresholding operation is only performed in the inference stage and the gradients do not pass through it during backpropagation.

Based on the mask $\boldsymbol{M}$, RN normalizes the input feature $\boldsymbol{F}$ and then performs a pixel-wise affine transformation. The affine parameters $\boldsymbol{\gamma} \in \mathbb{R}^{1 \times H \times W}$ and $\boldsymbol{\beta} \in \mathbb{R}^{1 \times H \times W}$ are obtained by convolution on the spatial response map $\boldsymbol{M_{sr}}$:

$$\boldsymbol{\gamma} = Conv(\boldsymbol{M_{sr}}), \boldsymbol{\beta} = Conv(\boldsymbol{M_{sr}}) \tag{16}$$

Note that the values of $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are expanded along the channel dimension in the affine transformation. The spatial response map $\boldsymbol{M_{sr}}$ has global spatial information. Convolution on it can learn a global representation, which boosts the fusion of corrupted and uncorrupted regions.

3.4 Network Architecture

EdgeConnect(EC) (Nazeri et al. 2019) consists of an edge generator and an image generator. The image generator is a simple yet effective network originally proposed by Johnson et al. (Johnson, Alahi, and Fei-Fei 2016). We use only
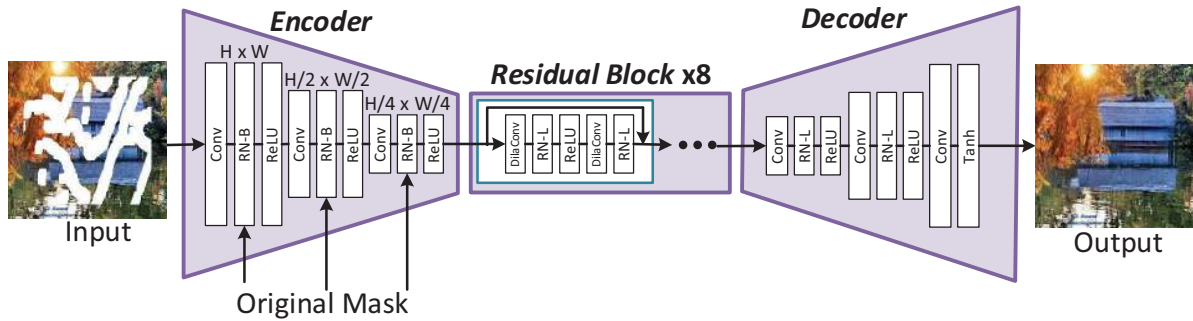
Figure 4: Illustration of our inpainting model.

the image generator as our backbone generator. We replace the original instance normalization (IN) of backbone generator to our two kinds of RN, RN-B and RN-L. Our generator architecture is shown in Figure 4. Based the instruction of Section 3.2 and 3.3, we apply RN-B in the early layers (encoder) of our generator and RN-L in the intermediate and later layers (the residual blocks and decoder). Note that the input mask of RN-B is sampled from the original inpainting mask while RN-L does not need an external input as it generates region masks internally. We apply the same discriminators (PatchGAN (Isola et al. 2017; Zhu et al. 2017)) and loss functions (reconstruction loss, adversarial loss, perceptual loss and style loss) of the original backbone model to our model[1].

4 Experiments

We first compare our method with current state-of-the-art methods. We then conduct ablation study to explore the properties of RN and visualize our methods. Finally, we generalize RN to some other state-of-the-art methods.

4.1 Experiment Setup

We evaluate our methods on Places2 (Zhou et al. 2017) and CelebA (Liu et al. 2015) datasets. We use two kinds of image masks: regular masks which are fixed square masks (occupying a quarter of the image) and irregular masks from (Liu et al. 2018). The irregular mask dataset contains 12000 irregular masks and the masked area in each mask occupies 0-60% of the total image size. Besides, the irregular dataset is grouped into six intervals according to the mask area, $i.e.$0-10%, 10-20%, 20-30%, 30-40%, 40-50% and 50-60%. Each interval has 2000 masks.

4.2 Comparison

We compare our method to four current state-of-the-art methods and the baseline.
- CA: Contextual Attention (Yu et al. 2018).
- PC: Partial Convolution (Liu et al. 2018).
- GC: Gated Convolution (Yu et al. 2019).
- EC: EdgeConnect (Nazeri et al. 2019).
- Baseline: the backbone network we used. The baseline model use instance normalization instead of RN.

[1]The codes are available at https://github.com/geekyutao/RN

	Mask	CA	PC*	GC	EC	baseline	Ours
PSNR↑	10-20%	24.45	28.02	26.65	27.46	27.28	**28.16**
	20-30%	21.14	24.90	24.79	24.53	24.35	**25.06**
	30-40%	19.16	22.45	**23.09**	22.52	22.33	22.94
	40-50%	17.81	20.86	**21.72**	20.90	20.96	21.21
	All	21.60	24.82	24.53	24.39	24.37	**25.10**
SSIM↑	10-20%	0.891	0.869	0.882	0.920	0.914	**0.926**
	20-30%	0.811	0.777	0.836	0.859	0.851	**0.868**
	30-40%	0.729	0.685	0.782	0.794	0.784	**0.804**
	40-50%	0.651	0.589	0.721	0.723	0.711	**0.734**
	All	0.767	0.724	0.807	0.814	0.806	**0.823**
$l_1(\%)$↓	10-20%	1.81	1.14	3.01	1.58	1.24	**1.10**
	20-30%	3.24	1.98	3.54	2.71	2.17	**1.96**
	30-40%	4.81	3.02	4.25	3.93	3.19	**2.90**
	40-50%	6.30	4.11	4.99	5.32	4.36	**4.00**
	All	4.21	2.80	3.79	2.83	2.95	**2.70**

Table 1: Quantitative results on Places2 with models: CA (Yu et al. 2018), PC (Liu et al. 2018), GC (Yu et al. 2019), EC (Nazeri et al. 2019), the baseline, and ours(RN). All masks $i.e.$ masks with 0-60% area. ↑ higher is better. ↓ lower is better. * the statistics are obtained from their paper.

Quantitative Comparisons We test all models on total validation data (36500 images) of Places2. We compare our model with CA, PC, GC, EC and the baseline. Three commonly used metrics are used: PSNR, SSIM (Wang et al. 2004) with window size 11, and l_1 loss. We give the results of quantitative comparisons in Table 1. The second column is the area of irregular masks at testing time. Note that the *All* in Table 1 represents using all irregular masks (0-60%) when testing. Our model surpasses all the comparing models on all three metrics. Compared to the baseline, our model improve PSNR by **0.73** dB and SSIM by **0.017**, and reduce l_1 loss (%) by **0.25** in the *All* case.

Qualitative Comparisons Figure 5 compares images generated by CA, PC, GC, EC, the baseline and ours. The first two rows of input images are taken from Places2 validation dataset and the last two rows are taken from CelebA validation dataset. In addition, the first three rows show the results in irregular mask case and the last row shows regular mask (fixed square mask in center) case. Our method achieves better subjective results, which benefits from RN-B's eliminating the impact of the mean and variance shifts on training, and RN-L's further boosting the reconstruction of corrupted regions.
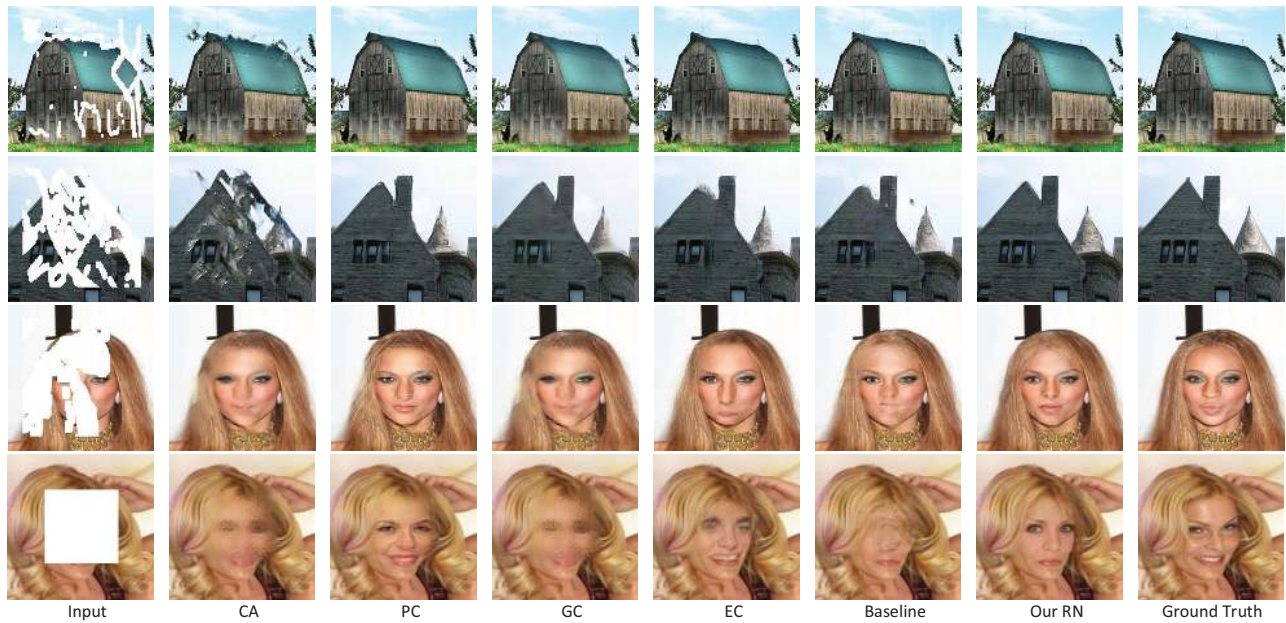
Figure 5: Qualitative results with CA (Yu et al. 2018), PC (Liu et al. 2018), GC (Yu et al. 2019), EC (Nazeri et al. 2019), the baseline, and our RN. The first two rows are the testing results on Places2 and the last two are on CelebA.

Arch.	Encoder	Res-blocks	Decoder	PSNR	SSIM	$l_1(\%)$
baseline	IN	IN	IN	24.37	0.806	2.95
1	**RN-B**	**IN**	**IN**	**24.88**	**0.814**	**2.77**
2	RN-B	RN-B	IN	24.41	0.810	2.90
3	RN-B	RN-B	RN-B	24.59	0.812	2.85
4	RN-B	RN-L	IN	25.02	**0.823**	2.71
5	**RN-B**	**RN-L**	**RN-L**	**25.10**	**0.823**	**2.70**
6	RN-L	RN-L	RN-L	24.53	0.812	2.86

Table 2: The influence of plugging location of RN-B and RN-L. The baseline uses inistance normalization (IN) in all three stages. The results are based on Places2.

	None	IN	BN	RN
PSNR	24.47	24.37	24.24	**25.10**
SSIM	0.811	0.806	0.806	**0.823**
$l_1(\%)$	2.91	2.95	2.98	**2.70**

Table 3: The final convergence results of different normalization methods on Places2. None means no normalization.

4.3 Ablation Study

RN and Architecture We first explore the source of gain for our methods and the best strategy to apply two kinds of RN: RN-B and RN-L. We conduct ablation experiments on the backbone generator, which has three stages: an encoder, followed by eight residual blocks and a decoder. We plug RN-B and RN-L in different stages and obtain six architectures (Arch.1-6) as shown in Table 2. The results in Table 2 show the effectiveness of our use of RN: apply RN-B in the early layers (encoder) to solve the mean and variance shifts caused by large-area uncorrupted regions; apply RN-L in the later layers to solve the the mean and variance shifts and boost the fusion of two kinds of regions. Arch.1 only applies RN-B in the encoder and achieves a significant
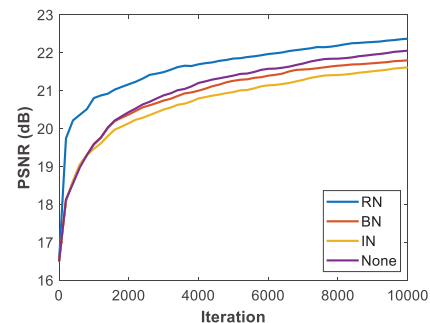

Figure 6: The PSNR results of different normalization methods in the first 10000 iterations on Places2. None means no normalization.

performance boost, which directly shows the RN-B's effectiveness. Arch.2 and 3 reduce the performance as RN-B can hardly obtain an accurate region mask in the latter layers of the network after passing through several convolutional layers. Arch.4 is beyond Arch.1 by adding RN-L in the middle residual blocks. Arch.5 (Our method) further improves the performance of Arch.4 by applying RN-L in both the residual blocks and the decoder. Note that Arch.6 uses RN-L to the encoder and its performance is reduced compared to Arch.5 ,since RN-L, a module of soft fusion, unavoidably mixing up information from corrupted and uncorrupted regions and washing away information from the uncorrupted regions. The above results verify the effectiveness of our use of RN-B and RN-L that we explain in Section 3.2 and 3.3.

Comparisons with Other Normalization Methods To verify our RN is more effective in training of the inpaint-
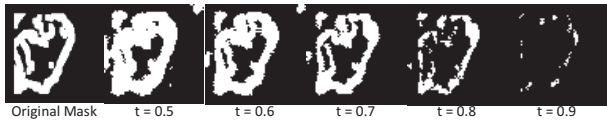
Figure 7: The generated mask with different threshold t of the first RN-L layer in the sixth residual block.

t	0.5	0.6	0.7	0.8	0.9
Places2	23.85	24.90	24.96	**25.10**	24.93
CelebA	27.36	27.92	28.45	**28.51**	23.73

Table 4: The PSNR results with different threshold t on Places2 and CelebA datasets.

ing model, we compare our RN with a none-normalization method and two full-spatial normalization methods, batch normalization (BN) and instance normalization (IN), based on the same backbone. We show the PSNR curves in the first 10000 iterations in Figure 6 and the final convergence results (about 225,000 iterations) in Table 3. The experiments are on Places2. Note that no normalization (None) is better than full-spatial normalization (IN and BN), and RN is better than no normalization by eliminating the mean and variance shifts and taking advantage of normalization technique at the same time.

Threshold of Learnable RN Threshold t is set in Learnable RN to generate a region mask from the spatial response map. The threshold affects the accuracy of the region mask and further affects the power of RN. We conduct a set of experiments to explore the best threshold. The PSNR results on Places2 and CelebA show that RN-L achieves the best results when threshold t equals to 0.8, as shown in Table 4. We show the generated mask of the first RN-L layer in the sixth residual block ($R6RN1$) as an example in Figure 7. The generated mask of $t = 0.8$ is likely to be the most accurate mask in this layer.

RN and Masked Area We explore the mask area's influence to RN. Based the theoretical analysis in Section 3.1, the mean and variance shifts become more severe as mask area increases. Our experiments on CelebA show that the advantage of our RN becomes more significant as the mask area increases, as shown in Table 5. We use l_1 loss to evaluate the results.

4.4 Visualization

We visualize some features of the inpainting network to verify our method. We show the changes of the spatial response and generated mask of RN-L as the network deepens in the top two rows of Figure 8. The mask changes in different layers as the fusion effect of passing through convolutional layers. RN-L can detect potentially corrupted regions consistently. From the last two rows of Figure 8 we can see: (1) the uncorrupted regions in the encoded feature are well preserved by using RN-B; (2) RN-L can distinguish between potentially different regions and generate a region mask; (3) gamma and beta maps in RN-L perform a pixel-level trans-

Mask	0-10%	10-20%	20-30%	30-40%	40-50%	50-60%
baseline	0.26	0.69	1.28	2.02	2.92	4.83
RN	**0.23**	**0.62**	**1.18**	**1.85**	**2.68**	**4.52**
Change	-0.03	-0.07	-0.10	-0.17	-0.24	-0.31

Table 5: The testing $l_1(\%)$ loss with different mask area on CelebA. RN's advantage becomes more significant as the mask area increases.

	CA	RN-CA	PC	RN-PC	GC	RN-GC
PSNR	21.60	**24.12**	24.82	**25.32**	24.53	**24.55**
SSIM	0.767	**0.842**	0.724	**0.829**	0.807	0.807
$l_1(\%)$	4.21	**3.17**	2.80	**2.61**	3.79	**3.75**

Table 6: The results of applying RN to different backbone networks: CA (Yu et al. 2018), PC (Liu et al. 2018) and GC (Yu et al. 2019). The results is based on Places2.
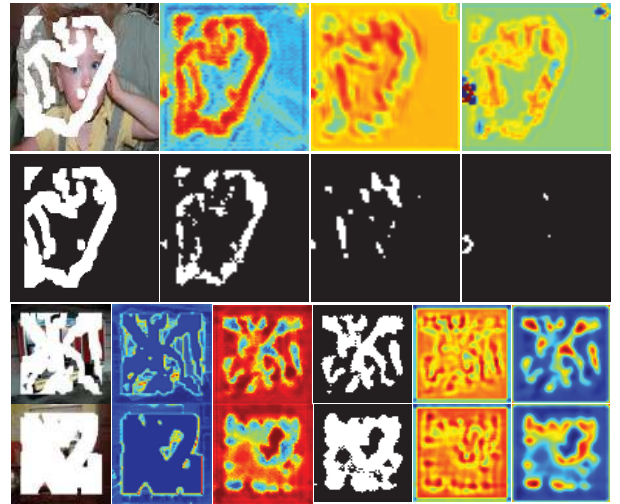

Figure 8: Visualization of our method. The top two rows are illustrated the changes of the spatial response and generated mask in different locations of the network: the first RN-L in the sixth residual block, the second RN-L in the seventh residual block and the second RN-L in the eighth residual block. In the last two rows, from left to right: input, encoder result, spatial response map, generated mask, gamma map and beta map of the first RN-L in the seventh residual block.

form on potentially corrupted and uncorrupted regions distinctively to help the fusion of them.

4.5 Generalization Experiments

RN-B and RN-L are plug-and-play modules in image inpainting networks. We generalize our RN (RN-B and RN-L) to some other backbone networks: CA, PC and GC. We apply RN-B to their early layers (encoder) and RN-L to the later layers. CA and GC are two-stage (coarse-to-fine) inpainting networks and the coarse result is the input of the refinement network. The corrupted and uncorrupted regions of the coarse result is typically not particularly obvious, thus we only apply RN to the coarse inpainting networks of CA and GC. The results on Places2 are shown in Table 6. The

RN-applied CA and PC achieve a significant performance boost by **2.52** and **0.5** dB PSNR respectively. The gain on GC is not very impressive. A possible reason is that gated convolution of GC greatly smoothes features which make RN-L hard to track potentially corrupted regions. Besides, GC's results are typically blurry as shown in Figure 5.

5 Conclusion

In this work, we investigate the impact of normalization on inpainting network and show that Region Normalization (RN) is more effective for image inpainting network, compared with existing full-spatial normalization. The proposed two kinds of RN are plug-and-play modules, which can be applied to other image inpainting networks conveniently. In the future, we will explore RN for other supervised vision tasks such as classification, detection and so on.

6 Acknowledgments

References

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Ballester, C.; Bertalmio, M.; Caselles, V.; Sapiro, G.; and Verdera, J. 2001. Filling-in by joint interpolation of vector fields and gray levels. *IEEE TIP* 10(8):1200–1211.

Barnes, C.; Shechtman, E.; Finkelstein, A.; and Goldman, D. B. 2009. Patchmatch: A randomized correspondence algorithm for structural image editing. In *ACM TOG*, volume 28, 24. ACM.

Bertalmio, M.; Sapiro, G.; Caselles, V.; and Ballester, C. 2000. Image inpainting. In *SIGGRAPH*, 417–424. ACM Press/Addison-Wesley Publishing Co.

Bertalmio, M.; Vese, L.; Sapiro, G.; and Osher, S. 2003. Simultaneous structure and texture image inpainting. *IEEE TIP* 12(8):882–889.

Darabi, S.; Shechtman, E.; Barnes, C.; Goldman, D. B.; and Sen, P. 2012. Image melding: Combining inconsistent images using patch-based synthesis. *ACM TOG* 31(4):82–1.

De Vries, H.; Strub, F.; Mary, J.; Larochelle, H.; Pietquin, O.; and Courville, A. C. 2017. Modulating early visual processing by language. In *NIPS*, 6594–6604.

Drori, I.; Cohen-Or, D.; and Yeshurun, H. 2003. Fragment-based image completion. In *ACM TOG*, volume 22, 303–312. ACM.

Dumoulin, V.; Shlens, J.; and Kudlur, M. 2016. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*.

Esedoglu, S., and Shen, J. 2002. Digital inpainting based on the mumford–shah–euler image model. *European Journal of Applied Mathematics* 13(4):353–370.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*, 2672–2680.

Huang, X., and Belongie, S. 2017. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 1501–1510.

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

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *CVPR*, 1125–1134.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 694–711. Springer.

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

Liu, G.; Reda, F. A.; Shih, K. J.; Wang, T.-C.; Tao, A.; and Catanzaro, B. 2018. Image inpainting for irregular holes using partial convolutions. In *ECCV*, 85–100.

Nazeri, K.; Ng, E.; Joseph, T.; Qureshi, F.; and Ebrahimi, M. 2019. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*.

Park, T.; Liu, M.-Y.; Wang, T.-C.; and Zhu, J.-Y. 2019. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2337–2346.

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

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2016. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; Simoncelli, E. P.; et al. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE TIP* 13(4):600–612.

Woo, S.; Park, J.; Lee, J.-Y.; and So Kweon, I. 2018. Cbam: Convolutional block attention module. In *ECCV*.

Wu, Y., and He, K. 2018. Group normalization. In *ECCV*, 3–19.

Xiong, W.; Yu, J.; Lin, Z.; Yang, J.; Lu, X.; Barnes, C.; and Luo, J. 2019. Foreground-aware image inpainting. In *CVPR*, 5840–5848.

Xu, Z., and Sun, J. 2010. Image inpainting by patch propagation using patch sparsity. *IEEE TIP* 19(5):1153–1165.

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

Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; and Huang, T. S. 2018. Generative image inpainting with contextual attention. In *CVPR*, 5505–5514.

Yu, J.; Lin, Z.; Yang, J.; Shen, X.; Lu, X.; and Huang, T. S. 2019. Free-form image inpainting with gated convolution. In *ICCV*, 4471–4480.

Zagoruyko, S., and Komodakis, N. 2016. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*.

Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Places: A 10 million image database for scene recognition. *IEEE TPAMI* 40(6):1452–1464.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2223–2232.