Generative Image Inpainting with Submanifold Alignment

Ang Li[*] , **Jianzhong Qi** , **Rui Zhang** , **Xingjun Ma**[*] and **Kotagiri Ramamohanarao**

The University of Melbourne

angl4@student.unimelb.edu.au, {jianzhong.qi, rui.zhang, xingjun.ma, kotagiri}@unimelb.edu.au

Abstract

Image inpainting aims at restoring missing regions of corrupted images, which has many applications such as image restoration and object removal. However, current GAN-based generative inpainting models do not explicitly exploit the structural or textural consistency between restored contents and their surrounding contexts. To address this limitation, we propose to enforce the alignment (or closeness) between the local data submanifolds (or subspaces) around restored images and those around the original (uncorrupted) images during the learning process of GAN-based inpainting models. We exploit Local Intrinsic Dimensionality (LID) to measure, in deep feature space, the alignment between data submanifolds learned by a GAN model and those of the original data, from a perspective of both images (denoted as iLID) and local patches (denoted as pLID) of images. We then apply iLID and pLID as regularizations for GAN-based inpainting models to encourage two levels of submanifold alignment: 1) an image-level alignment for improving structural consistency, and 2) a patch-level alignment for improving textural details. Experimental results on four benchmark datasets show that our proposed model can generate more accurate results than state-of-the-art models.

1 Introduction

Given a corrupted image where part of the image is missing, image inpainting aims to synthesize plausible contents that are coherent with non-missing regions. Figure 1 illustrates such an example, where Figure 1a and 1b show two original (uncorrupted) images and their corrupted versions with missing regions respectively. The aim is to restore the missing regions of corrupted images in Figure 1b, such that the restored regions contain contents that make the whole image look natural and undamaged (high structural and textural consistency with the original images in Figure 1a). With the help of image inpainting, applications such as restoring damaged images or removing blocking contents from images can be realized.

---

[*]Corresponding authors.

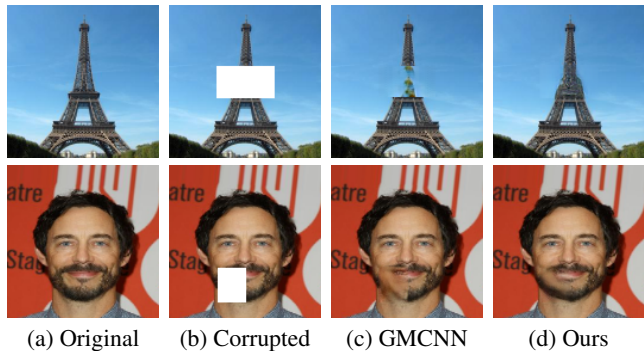|			
(a) Original	(b) Corrupted	(c) GMCNN	(d) Ours

Figure 1: Comparison between the state-of-the-art GMCNN and our proposed model.

Generative Adversarial Networks (GANs) are powerful models for image generation [Goodfellow *et al.*, 2014; Radford *et al.*, 2016]. From the perspective of GANs, image inpainting can be viewed as a conditional image generation task given the context of uncorrupted regions [Pathak *et al.*, 2016; Iizuka *et al.*, 2017; Yang *et al.*, 2017; Yeh *et al.*, 2017; Yu *et al.*, 2018; Wang *et al.*, 2018]. A representative GAN-based inpainting model is Context Encoder [Pathak *et al.*, 2016], where a convolutional encoder-decoder network is trained with the combination of a reconstruction loss and an adversarial loss [Goodfellow *et al.*, 2014]. The reconstruction loss guides to recover the overall coarse structure of the missing region, while the adversarial loss guides to choose a specific distribution from the results to promote natural-looking patterns. Lately, Wang *et al.* [2018] propose a generative multi-column neural network (GMCNN) for better restoration.

In spite of the encouraging results, the restored images by current GAN-based inpainting models may lead to *structural or textural inconsistency* compared with the original images. Specifically, current inpainting models may significantly alter the structure or the texture of the objects involved in the missing regions, and therefore the restored contents visually mismatch their surrounding undamaged contents. For example, in Figure 1c, the restored regions by the state-of-the-art inpainting model GMCNN [Wang *et al.*, 2018] exhibit segmentation structure misalignment (part of the tower is missing) in the top image and texture mismatch (inconsistent with respect to 'beard' pattern) in the bottom image. This problem

can be ascribed to the following two limitations.

First, GAN-based formulation has the ill-posed nature: the one-to-many mapping relationships between a missing region and its possible restorations. This can lead to sub-optimal inpainting results as in the above examples, especially when the inpainting model was not properly regularized for structural/textural consistency. The commonly used loss combination of a reconstruction loss and an adversarial loss, however, fails to address this limitation. The reconstruction loss mainly focuses on minimizing pixel-wise difference (very low-level), while the adversarial loss determines the overall similarity to real images (very high-level). Intuitively, mid-level (between the pixel and the overall realness levels) regularizations that can provide more explicit emphases on both structural and textural consistencies are needed for natural restorations.

Second, most existing inpainting models [Yang *et al.*, 2017; Yu *et al.*, 2018] adopt a two-stage "coarse-to-fine" inpainting process. Specifically, these models generate a coarse result at the first stage (which also suffers from the first limitation), then refine the coarse result at the second stage by replacing patches in the restored regions with their nearest neighbors found in the original patches. Due to the cosine-similarity based searching of nearest patches, these models tend to produce repetitive patterns. Moreover, when structures in missing regions are originally different from those in the background, these models may significantly alter or distort the structure of the restored contents and end up with unrealistic restorations, for example in the top image of Figure 1c, filling missing part of the 'tower' with 'sky' background.

To address the two limitations, we propose to enforce the alignment (or closeness) between the local data submanifolds (or subspaces) around restored images and those around the original images, during the learning process of GAN-based inpainting models. Our intuition is that, a restored image or patch of the image will look more natural if the image or the patch is drawn from a data submanifold that is closely surrounding the original image or patch, that is, data submanifolds around the restored images/patches align well with those around the original images/patches. In this paper, we adapt an expansion-based dimensionality measure called Local Intrinsic Dimensionality (LID) to characterize the dimensional property of the local data submanifold (in deep feature space) around a reference image/patch. We further exploit LID to measure the *submanifold alignment* from a perspective of either images or local patches. This allows us to develop two different levels of submanifold alignment regularizations to improve GAN-based inpainting models: 1) an image-level submanifold alignment (iLID) for structural consistency, and 2) a patch-level submanifold alignment (pLID) for improving textural details. In summary, our main contributions are:

- We propose and generalize the use of Local Intrinsic Dimensionality (LID) in image inpainting to measure how closely submanifolds around restored images/patches align with those around the original images/patches.

- With the generalized LID measure, we develop two submanifold alignment regularizations for GAN-based inpainting models to improve structural consistency and textural details.

- Experiments show that our model can effectively reduce structural/textural inconsistency and achieve more accurate inpainting compared with state-of-the-art models.

2 Related Work

Traditional image inpainting models are mostly based on patch matching [Barnes *et al.*, 2009; Bertalmio *et al.*, 2003] or texture synthesis [Efros and Leung, 1999; Efros and Freeman, 2001]. They often suffer from low generation quality, especially when dealing with large arbitrary missing regions. Recently, deep learning and GAN-based approaches have been employed to produce more promising inpainting results [Pathak *et al.*, 2016; Yang *et al.*, 2017; Yeh *et al.*, 2017; Iizuka *et al.*, 2017; Yu *et al.*, 2018; Liu *et al.*, 2018; Song *et al.*, 2018; Zhang *et al.*, 2018; Wang *et al.*, 2018]. Pathak *et al.*[2016] first propose the *Context Encoder* (CE) model that has an encoder-decoder CNN structure. They train CE with the combination of a reconstruction loss and an adversarial loss [Goodfellow *et al.*, 2014]. Later models apply post-processing on images produced by encoder-decoder models to further improve their quality. Yang *et al.* [2017] propose one such model that adopts CE at its first stage, while at a second stage, they refine the inpainting results by propagating surrounding texture information. Iizuka *et al.* [2017] propose to use both global and local discriminators at the first stage, and Poisson blending at the second stage. Yu *et al.* [2018] propose a refinement network for post-processing based on contextual attention. These models mainly focus on enhancing the resolution of inpainted images, while ignoring the structural or textural consistency between the restored contents and their surrounding contexts. These models are all based on an encoder-decoder model for generating initial results, which can easily suffer from the ill-posed one-to-many ambiguity. This may produce suboptimal intermediate results that limit the effectiveness of post-processing at a later stage. Lately, Wang *et al.* [2018] propose a generative multi-column neural network (GMCNN) for better restoration of global structures, however, as we show in Figure 1c, it still suffers from the structural or textural inconsistency problem.

3 Dimensional Characterization of Local Data Submanifolds

We start with a brief introduction of the Local Intrinsic Dimensionality (LID) [Houle, 2017] measure for assessing the dimensional properties of local data submanifolds/subspaces.

3.1 Local Intrinsic Dimensionality

LID measures the rate of growth (*e.g.* expansion rate) in the number of data points encountered as the distance from a reference point increases. Intuitively, in Euclidean space, the volume of an D-dimensional ball grows proportionally to r^D when its size is scaled by a factor of r. From the above rate of volume growth with distance, the dimension D can be deduced from two volume measurements as:

$$V_2/V_1 = (r_2/r_1)^D \Rightarrow D = \ln(V_2/V_1)/\ln(r_2/r_1). \quad (1)$$

Transferring the concept of expansion dimension from the Euclidean space to the statistical setting of local distance distributions, the notion of r becomes the distance from a reference sample x to its neighbours and the ball volume is replaced by the cumulative distribution function of the distance. This leads to the formal definition of LID [Houle, 2017].

Definition 1 (Local Intrinsic Dimensionality)
Given a data sample $x \in X$, let $r > 0$ be a random variable denoting the distance from x to other data samples. If the cumulative distribution function $F(r)$ is positive and continuously differentiable at distance $r > 0$, the LID of x at distance r is given by:

$$\mathrm{LID}_F(r) \triangleq \lim_{\epsilon \to 0} \frac{\ln\left(F((1+\epsilon)r)/F(r)\right)}{\ln((1+\epsilon)r/r)} = \frac{rF'(r)}{F(r)}, \quad (2)$$

whenever the limit exists. The LID at x is in turn defined as the limit when the radius $r \to 0$:

$$\mathrm{LID}_F = \lim_{r \to 0} \mathrm{LID}_F(r). \quad (3)$$

The last equality of Equation (2) follows by applying L'Hôpital's rule to the limits. LID_F describes the relative rate at which the cumulative distance function $F(r)$ increases as the distance r increases. In the ideal case where the data in the vicinity of x are distributed uniformly within a local submanifold, LID_F equals the dimension of the submanifold. In more general cases, LID provides a rough indication of the dimension of the submanifold containing x that would best fit the data distribution in the vicinity of x.

3.2 Estimation of LID

There already exists several estimators of LID [Amsaleg *et al.*, 2015; 2018; Levina and Bickel, 2005], among which the Maximum Likelihood Estimator (MLE) shows the best trade-off between statistical efficiency and complexity:

$$\mathrm{LID}(x; X) = -\left(\frac{1}{k}\sum_{i=1}^{k}\ln\frac{r_i(x; X)}{r_{max}(x; X)}\right)^{-1}. \quad (4)$$

where k is the neighborhood size, $r_i(x; X)$ is the distance from x to its i-th nearest neighbor in $X \setminus \{x\}$, and $r_{max}(x; X)$ denotes the maximum distance within the neighborhood (which by convention can be $r_k(x; X)$).

LID has recently been used in adversarial detection [Ma *et al.*, 2018a] to characterize adversarial subspaces of deep networks. It has also been applied to investigate the dimensionality of data subspaces in the presence of noisy labels [Ma *et al.*, 2018b]. In a recent work [Barua *et al.*, 2019], LID was exploited to measure the quality of GANs in terms of the degree to which manifolds of real data distribution and generated data distribution coincide with each other. Inspired by these studies, in this paper, we develop the use of LID in image inpainting to encourage submanifold alignment for GAN-based generative inpainting models.

4 Proposed Inpainting Model with Submanifold Alignment

Given an input (corrupted) image $x \in X$ with missing regions, an image inpainting model aims to restore the missing
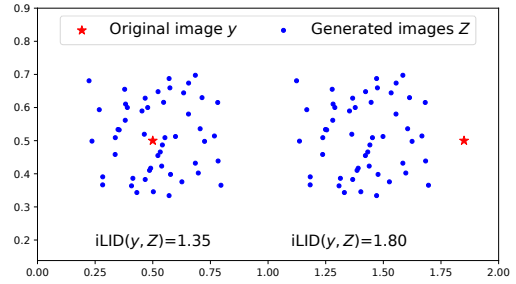

Figure 2: This example shows how $\mathrm{iLID}(y, Z)$ can reflect the closeness of the data submanifold of restored images Z (blue points) with respect to an original image y (red star).

regions so that the output (restored) image $z \in Z$ matches the ground truth (original) image $y \in Y$. Following previous works, we adopt GANs to generate the output image z. To address the structural and textural inconsistencies exist in current GAN-based inpainting models, we propose to enforce the restored images or local patches to be drawn from data submanifolds that align closely with submanifolds of the original images or patches. To achieve this, we develop the LID characterization of local data submanifolds to derive two levels of submanifold alignments: image-level and patch-level.

4.1 Image-level Submanifold Alignment

We propose *Image-level Local Intrinsic Dimensionality* (iLID) to measure how close data submanifolds around restored images are to those around the original images. For an original image, we define the iLID of its local submanifold by its neighbors found in GAN generated samples:

$$\mathrm{iLID}(y; Z) = -\left(\frac{1}{k_I}\sum_{i=1}^{k_I}\log\frac{r_i(\phi(y), \phi(Z))}{r_{max}(\phi(y), \phi(Z))}\right)^{-1}, \quad (5)$$

where, transformation ϕ represents the L_{th} layer of a pretrained network mapping raw images into deep feature space, $r_i(\phi(y), \phi(Z))$ is the L_2 distance of $\phi(y)$ to its i-th nearest neighbor $\phi(z_i)$ in $\phi(Z)$, and k_I is the neighborhood size.

Figure 2 illustrates a toy example showing how $\mathrm{iLID}(y; Z)$ works. In the left scenario, GAN learns a data submanifold (with blue points indicating samples Z drawn from this submanifold) that is closely surrounding the original image y and the $\mathrm{iLID}(y; Z)$ score is low, while in the right scenario, as the GAN-learned data submanifold drifts away from y the $\mathrm{iLID}(y; Z)$ score increases from 1.53 to 3.78. Note that, from the original LID perspective, the right scenario can be interpreted as a sample y lies in a higher dimensional space that is off the normal data submanifold. Revisit Eq. (5), assuming $r_{max} > r_i$, as the distance between y and its nearest neighbors in Z increases by $d > 0$, then the term inside of the log function becomes $(r_i + d)/(r_{max} + d)$. Since,

$$\frac{r_i + d}{r_{max} + d} - \frac{r_i}{r_{max}} = \frac{r_{max} - r_i}{r_{max}(r_{max}/d + 1)}, \quad (6)$$

$\mathrm{iLID}(y; Z)$ will increase with the increase of d, reflecting the alignment/closeness drift of local data submanifold from y.

Taking the expectation of iLID$(y; Z)$ over all the original samples in Y gives us the regularization for image-level submanifold alignment:

$$\mathcal{L}_{\text{iLID}}(Y; Z) = \mathbb{E}_{y \in Y} \text{ iLID}(y; Z). \qquad (7)$$

Low $\mathcal{L}_{\text{iLID}}(Y; Z)$ scores indicate low average spatial distance between the elements of Y and their neighbor sets in Z. $\mathcal{L}_{\text{iLID}}(Y; Z)$ provides a local view of how well data submanifolds learned by a GAN model align with the original data submanifolds. By optimizing $\mathcal{L}_{\text{iLID}}(Y; Z)$, we can encourage a GAN generator to draw restorations from data submanifolds that are closely surrounding the original images, thus can avoid structural inconsistency between the restored image and the original image.

As computing neighborhoods for each $y \in Y$ with respect to large sets Z can be prohibitively expensive, we use a batch-based estimation of LID: for one batch during training, we use the GAN generated samples as the Z for that batch to compute iLID score for each original sample in the batch. In other words, the $\mathcal{L}_{\text{iLID}}(Y; Z)$ is efficiently defined over a batch of samples (original or restored).

4.2 Patch-level Submanifold Alignment

To avoid local textural inconsistency, we further propose the *Patch-level Local Intrinsic Dimensionality* (pLID) based on feature patches in the deep feature space (rather than the raw pixel space). Feature patches were extracted on the feature map obtained by the same transformation ϕ as used for image-level submanifold alignment. Specifically, for an original image $y \in Y$ and its corresponding restored image $z \in Z$ with restored region z^o, we extract one set P of 3×3 feature patches from the entire feature map $\phi(y)$, and one set Q of 3×3 feature patches only from part of the feature map $\phi(z)$ that is associated with region z^o. Following similar formulation of iLID, we define pLID as:

$$\text{pLID}(p; Q) = -\left(\frac{1}{k_P} \sum_{i=1}^{k_P} \log \frac{r_i(p, Q)}{r_{max}(p, Q)} \right)^{-1}, \quad (8)$$

where, $p \in P$ is a patch in P and k_P is the neighborhood size. Then, the patch-level submanifold alignment regularization can be defined as:

$$\mathcal{L}_{\text{pLID}}(Y; Z) = \mathbb{E}_{y \in Y} \mathbb{E}_{p \in P} \text{ pLID}(p; Q). \qquad (9)$$

Optimizing $\mathcal{L}_{\text{pLID}}(Y; Z)$ enforces the GAN model to draw patches from data submanifolds that are closely surrounding the original patches, which helps improve texture consistency. Specifically, $\mathcal{L}_{\text{pLID}}(p; Q)$ encourages p to receive reference from different neural patches in Q, thus can avoid repetitive patterns and maintain variety in local texture details. Similar to $\mathcal{L}_{\text{iLID}}(Y; Z)$, we define the $\mathcal{L}_{\text{pLID}}(Y; Z)$ over a batch of original or restored samples during training.

4.3 The Overall Training Loss

The overall training loss of our model is a combination of an adversarial loss, a reconstruction loss and the two proposed regularizations:

$$\mathcal{L} = \lambda_I \mathcal{L}_{\text{iLID}} + \lambda_P \mathcal{L}_{\text{pLID}} + \lambda_A \mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{rec}}, \qquad (10)$$

where $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{rec}}$ denote the adversarial loss and reconstruction loss respectively, and parameters λ_I, λ_P and λ_A control the tradeoff between different components of the loss. Following recent works [Liu *et al.*, 2018; Yu *et al.*, 2018], we adopt the improved Wasserstein GAN [Gulrajani *et al.*, 2017] as our adversarial loss:

$$\mathcal{L}_{\text{adv}} = -\mathbb{E}_{x \sim \mathbb{P}_x}[D(G(x))] + \lambda_{gp} \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\| - 1)^2],$$
$$(11)$$

where G and D denote the generator and the discriminator of our model respectively, $x \in X$ is an input (corrupted) sample and $z = G(x)$ is the GAN generated sample. The restored image for x is $\hat{x} = tx + (1-t)G(x)$ with the mask $t \in \{0, 1\}$ having zeros at the missing pixels and ones at the non-missing pixels, and we use the L_2 reconstruction loss:

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{x \sim \mathbb{P}_x, y \sim \mathbb{P}_y} \|\hat{x} - y\|_2. \qquad (12)$$

5 Experiments

We compare with four state-of-the-art inpainting models: **CE** [Pathak *et al.*, 2016], **GL** [Iizuka *et al.*, 2017], **GntIpt** [Yu *et al.*, 2018] and **GMCNN** [Wang *et al.*, 2018]. We apply the proposed training strategy on both CE and GL GAN-architectures, and denote their improved versions as **CE+LID** and **GL+LID** respectively. We evaluate all models on four benchmark datasets: Paris StreetView [Doersch *et al.*, 2012], CelebA [Liu *et al.*, 2015], Places2 [Zhou *et al.*, 2017] and ImageNet [Russakovsky *et al.*, 2015]. No pre-processing or post-processing is applied in all experiments.

We set λ_A to 0.01 as in [Yu *et al.*, 2018; Iizuka *et al.*, 2017]. Based on our analysis in Section 5.3, we empirically set $\lambda_I = 0.01$, $\lambda_P = 0.1$, $k_I = 8$ and $k_P = 5$ for our models. All training images are resized and cropped to 256×256. We choose *conv4_2* of VGG19 [Simonyan and Zisserman, 2014] as the transformation ϕ. For all the tested models, we mask an image with a rectangular region that has a random location and a random size (ranging from 40×40 to 160×160). The size of a training batch is 64. The training takes 16 hours on Paris StreetView, 20 hours on CelebA and one day on both Places2 and ImageNet using an Nvidia GTX 1080Ti GPU.

5.1 Quantitative Comparisons

We compute the *Peak Signal-to-Noise Ratio* (PSNR) and the *Structural Similarity* (SSIM) between a restored image and its ground truth for quantitative comparison as in [Yu *et al.*, 2018; Wang *et al.*, 2018]. As shown in Table 1, both of our proposed models CE+LID and GL+LID achieve better or comparable results to existing models across the four datasets, which verifies the effectiveness of submanifold alignment on different inpainting models. Since GL+LID demonstrates more promising results than CE+LID, we only focus on the GL+LID model in the following experiments.

5.2 Qualitative Comparisons

We show sample images in this section to highlight the performance difference between our models and the four state-of-the-art models on the selected four datasets. As shown in Figure 3, CE [Pathak *et al.*, 2016] can generate overall reasonable contents though the contents may be coarse and semantically inconsistent with surrounding contexts. For example,

Model	Paris StreetView		CelebA		Places2		ImageNet	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CE [Pathak *et al.*, 2016]	22.92	0.858	21.78	0.923	18.72	0.843	20.43	0.787
GL [Iizuka *et al.*, 2017]	23.64	0.860	23.19	0.936	19.09	0.836	21.85	0.853
GntIpt [Yu *et al.*, 2018]	24.12	0.862	23.80	0.940	20.38	0.855	22.82	**0.879**
GMCNN [Wang *et al.*, 2018]	23.82	0.857	24.46	0.944	20.62	0.851	22.16	0.864
Ours (CE+LID)	23.78	**0.873**	24.05	0.947	20.17	0.848	22.11	0.860
Ours (GL+LID)	**24.33**	0.867	**25.56**	**0.953**	**21.02**	**0.864**	**23.50**	0.875

Table 1: Quantitative comparisons on Paris Street View, CelebA, Places2 and ImageNet.

(a) Original　　(b) Corrupted　　(c) CE　　(d) GL　　(e) GntIpt　　(f) GMCNN　　(g) Ours (GL+LID)

Figure 3: Qualitative comparisons on Paris StreetView, CelebA, Places2 and ImageNet.

the input image of the first row shows an building that has a part of the window missing. CE can only generate coarse content with incomplete structures of the window and blurry details. This is due to the limitation of the original GAN model that only provides overall high-level guidance (whether the inpaining output is generally similar to the ground truth image) and lacks specific regularization. With the help of global and local discriminators, GL [Iizuka *et al.*, 2017] can generate better details than CE but it still suffers from generating corrupted structures and blurry texture details. Although GntIpt [Yu *et al.*, 2018] can restore sharp details, it can lead to structural or textural inconsistency so that the restored contents may be contradictory with the surrounding contexts. Since the first stage of GntIpt resembles the overall structure of GL, the coarse outputs from the first stage of GntIpt also have similar problems with GL. Moreover, the second stage of GntIpt (finding the closest original patch from the surroundings with cosine similarity for the restored content) tends to easily attach simple and repeating patterns onto the restored content, thus even worsens the above mentioned problem. Specifically

in the first image, it fills the missing region with wall pattern ignoring the remaining window part at the bottom. GMCNN [Wang *et al.*, 2018] proposes an improved similarity measure to encourage diversity when finding the patch match, however it only insists on single patch diversity and thus causes unexpected artifacts and noises in the restored content. In contrast, our model can generate more realistic results with better structural/textural consistency compared with the four state-of-the-art models.

5.3 Ablation Study

Parameter Analysis on iLID and pLID Regularizations

We further conduct ablation experiments for our model on Places2. Specifically, we investigate different combinations of iLID regularization and pLID regularization in the loss function by assigning λ_I and λ_P with different values in $[0.001, 1]$. An example of qualitative comparisons with different combinations of λ_I and λ_P is shown in Figure 4. When λ_I and λ_P are both set to 0, the restored image encounters both structural and textural inconsistency problem with part
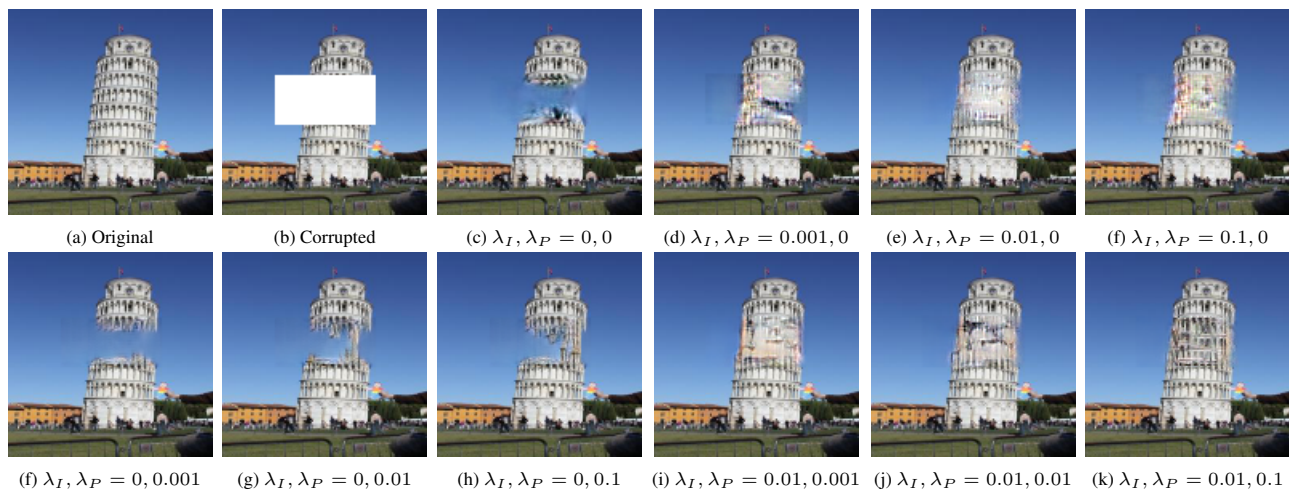
(a) Original　(b) Corrupted　(c) $\lambda_I, \lambda_P = 0, 0$　(d) $\lambda_I, \lambda_P = 0.001, 0$　(e) $\lambda_I, \lambda_P = 0.01, 0$　(f) $\lambda_I, \lambda_P = 0.1, 0$

(f) $\lambda_I, \lambda_P = 0, 0.001$　(g) $\lambda_I, \lambda_P = 0, 0.01$　(h) $\lambda_I, \lambda_P = 0, 0.1$　(i) $\lambda_I, \lambda_P = 0.01, 0.001$　(j) $\lambda_I, \lambda_P = 0.01, 0.01$　(k) $\lambda_I, \lambda_P = 0.01, 0.1$

Figure 4: Qualitative comparisons of results with different combinations of λ_I and λ_P.

$\lambda_I = 0$			
λ_P	0.001	0.01	0.1	1
PSNR	20.42	20.69	**20.94**	20.46
SSIM	0.822	0.837	**0.848**	0.820
$\lambda_P = 0$			
λ_I	0.001	0.01	0.1	1
PSNR	20.51	**20.86**	20.80	20.54
SSIM	0.833	**0.853**	0.844	0.830
$\lambda_I = 0.01$			
λ_P	0.001	0.01	0.1	1
PSNR	20.81	21.08	**21.21**	20.84
SSIM	0.848	0.856	**0.864**	0.843
$\lambda_P = 0.1$			
λ_I	0.001	0.01	0.1	1
PSNR	20.66	**21.21**	21.02	20.75
SSIM	0.837	**0.864**	0.858	0.839

Table 2: Quantitative comparisons with different combinations of λ_I and λ_P on Places2.

(a) Corrupted　(b) $k_P = 1$　(c) $k_P = 5$　(d) $k_P = 9$

Figure 5: Qualitative comparisons on different k_P values.

of the building missing and filled with background (sky) pattern. If we assign λ_I with non-zero values and remain λ_P as 0, the model becomes able to restore the overall structure of the building even if the texture details are noisy and blurry. When we assign λ_P with non-zero values and remain λ_I as 0, the restored results have sharper and more diverse texture details on local patch level, but they fail to be consistent with the structure of original image on overall image level. When we both assign λ_I and λ_P with non-zero values, we notice that the restored images are consistent on not only overall structure but also texture details with the surrounding context. From the visual results, we see that iLID regulariza-

tion can increase structural consistency of restored contents, while pLID regularization focuses on patch-level alignment and thus benefits textural consistency. Quantitative comparisons with 4 groups of parameter setting are given in Table 2. We see that both iLID regularization and pLID regularization contribute to the model performance when using them alone. Moreover, the collaboration of these two different regularizations can achieve complementary effects to improve inpainting quality. We therefore empirically set $\lambda_I = 0.01$ and $\lambda_P = 0.1$ based on the ablation experiments.

The Effect of Different Neighborhood Sizes

We also investigate the effect of different neighborhood sizes on iLID (k_I) and pLID (k_P) regularizations. For iLID regularization, different values of k_I achieve similar results and empirically $k_I = 5 \sim 10$ strikes a better balance. For pLID regularization, we find that assigning k_P with different values can bring variable texture diversity for results. Figure 5 shows that $k_P = 5$ leads to more accurate texture details.

6 Conclusion

We studied the structural/textural inconsistency problem in image inpainting. To address this limitation, we propose to enforce the alignment between the local data submanifolds around restored images and those around the original images. Our proposed inpainting model utilizes a combination of two LID-based regularizations: an image-level alignment regularization (with iLID) and a patch-level alignment regularization (with pLID) during the training process. The experimental results confirm that our model achieves more accurate inpainting compared with the state-of-the-art models on multiple datasets. We note that our enhancement can be applied to any GAN-based inpainting model, and further analysis in this direction can be interesting future work.

Acknowledgements

References

[Amsaleg *et al.*, 2015] Laurent Amsaleg, Oussama Chelly, Teddy Furon, Stéphane Girard, Michael E. Houle, Kenichi Kawarabayashi, and Michael Nett. Estimating local intrinsic dimensionality. In *SIGKDD*, 2015.

[Amsaleg *et al.*, 2018] Laurent Amsaleg, Oussama Chelly, Teddy Furon, Stéphane Girard, Michael E. Houle, Ken Ichi Kawarabayashi, and Michael Nett. Extreme-value-theoretic estimation of local intrinsic dimensionality. *Data Mining & Knowledge Discovery*, 2018.

[Barnes *et al.*, 2009] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 2009.

[Barua *et al.*, 2019] Sukarna Barua, Xingjun Ma, Sarah Monazam Erfani, Michael E Houle, and James Bailey. Quality evaluation of gans using cross local intrinsic dimensionality. *arXiv preprint arXiv:1905.00643*, 2019.

[Bertalmio *et al.*, 2003] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 2003.

[Doersch *et al.*, 2012] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei Efros. What makes paris look like paris? *ACM Transactions on Graphics*, 2012.

[Efros and Freeman, 2001] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, 2001.

[Efros and Leung, 1999] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *ICCV*, 1999.

[Goodfellow *et al.*, 2014] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[Gulrajani *et al.*, 2017] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *NIPS*, 2017.

[Houle, 2017] Michael E. Houle. Local intrinsic dimensionality i: an extreme-value-theoretic foundation for similarity applications. In *SISAP*, 2017.

[Iizuka *et al.*, 2017] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 2017.

[Levina and Bickel, 2005] Elizaveta Levina and Peter J. Bickel. Maximum likelihood estimation of intrinsic dimension. In *NIPS*, 2005.

[Liu *et al.*, 2015] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[Liu *et al.*, 2018] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, 2018.

[Ma *et al.*, 2018a] Xingjun Ma, Li Bo, Yisen Wang, Sarah M. Erfani, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *ICLR*, 2018.

[Ma *et al.*, 2018b] Xingjun Ma, Yisen Wang, Michael E. Houle, Shuo Zhou, Sarah M. Erfani, Shu-Tao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. *ICML*, 2018.

[Pathak *et al.*, 2016] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[Radford *et al.*, 2016] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *ICLR*, 2016.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014.

[Song *et al.*, 2018] Yuhang Song, Chao Yang, Zhe Lin, Xiaofeng Liu, Qin Huang, Hao Li, and C-C Jay Kuo. Contextual-based image inpainting: Infer, match, and translate. In *ECCV*, 2018.

[Wang *et al.*, 2018] Yi Wang, Xin Tao, Xiaojuan Qi, Xiaoyong Shen, and Jiaya Jia. Image inpainting via generative multi-column convolutional neural networks. In *NeurIPS*, 2018.

[Yang *et al.*, 2017] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *CVPR*, 2017.

[Yeh *et al.*, 2017] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *CVPR*, 2017.

[Yu *et al.*, 2018] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *CVPR*, 2018.

[Zhang *et al.*, 2018] Haoran Zhang, Zhenzhen Hu, Changzhi Luo, Wangmeng Zuo, and Meng Wang. Semantic image inpainting with progressive generative networks. In *ACMMM*, 2018.

[Zhou *et al.*, 2017] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.