RESEARCH-ARTICLE

# Stacked Semantically-Guided Learning for Image De-distortion

**HUIYUAN FU**, Beijing University of Posts and Telecommunications, Beijing, Beijing, China

**CHANGHAO TIAN**, Beijing University of Posts and Telecommunications, Beijing, Beijing, China

**XIN WANG**, Stony Brook University, Stony Brook, NY, United States

**HUADONG MA**, Beijing University of Posts and Telecommunications, Beijing, Beijing, China

**Open Access Support** provided by:

**Stony Brook University**

**Beijing University of Posts and Telecommunications**



**Published:** 17 October 2021

**Citation in BibTeX format**

MM '21: ACM Multimedia Conference
*October 20 - 24, 2021*
*Virtual Event, China*

**Conference Sponsors:**
SIGMM


https://doi.org/10.1145/3474085.3475608
ISBN: 9781450386517

# Stacked Semantically-Guided Learning for Image De-distortion

Huiyuan Fu
Beijing University of Posts and Telecommunications
Beijing, China
fhy@bupt.edu.cn

Changhao Tian
Beijing University of Posts and Telecommunications
Beijing, China
chaunhewietian@bupt.edu.cn

Xin Wang
Stony Brook University
New York, USA
x.wang@stonybrook.edu

Huadong Ma
Beijing University of Posts and Telecommunications
Beijing, China
mhd@bupt.edu.cn

## ABSTRACT

Image de-distortion is very important because distortions will degrade the image quality significantly. It can benefit many computational visual media applications that are primarily designed for high-quality images. In order to address this challenging issue, we propose a stacked semantically-guided network, which is the first try on this task. It can capture and restore the distortions around the humans and the adjacent background effectively with the stacked network architecture and the semantically-guided scheme. In addition, a discriminative restoration loss function is proposed to recover different distorted regions in the images discriminatively. As another important effort, we construct a large-scale dataset for image de-distortion. Extensive qualitative and quantitative experiments show that our proposed method achieves a superior performance compared with the state-of-the-art approaches.

## CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; **Machine learning algorithms**; **Image processing**.

## KEYWORDS

image de-distortion; stacked; semantically; generative adversarial networks

**ACM Reference Format:**
Huiyuan Fu, Changhao Tian, Xin Wang, and Huadong Ma. 2021. Stacked Semantically-Guided Learning for Image De-distortion. In *Proceedings of the 29th ACM Int'l Conference on Multimedia (MM '21), Oct. 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475608

## 1 INTRODUCTION

Image de-distortion is very important for many computation-based visual media applications, such as photo modification, portrait revision, and intelligent body beautification. Distorted images affect the overall structure significantly, and ruin the beauty of the originals to a certain extent. If the distorted regions can be found and removed or alleviated after the image de-distortion, images can become more natural.

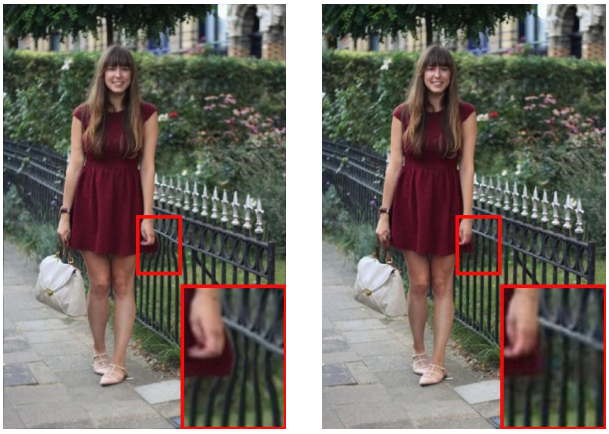

**Figure 1: An example result of our proposed stacked semantically-guided network. Left: The raw distorted image. Right: The output image of our method.**

Recently, deep neural networks have achieved impressive performance in image restoration [4, 7], image inpainting [21, 23], and image super-resolution [8, 20]. These promising results motivate us to study image de-distortion with the deep neural networks. Specifically, in light of the remarkable progress in multiple image-to-image translation with Generative Adversarial Networks (GANs) [10, 24], we will take full advantage of GANs for image de-distortion.

Despite the importance, we have not found an open study for image de-distortion. To the best of our knowledge, we are the first researchers to dedicate for this task. For high-quality image de-distortion, we propose a Stacked Semantically-Guided Network (SSGN) that effectively capture and restore the distortions around the humans and the adjacent background. Taking advantage of the fact that distorted regions are more perceptible in low resolution images, we design a stacked network architecture to realize the gradual recovery from the low resolution to the high resolution. In addition, with the help of a semantically-guided scheme, we can pay more attention to the surrounding areas of a human, to enhance the perception of the network around the human body. What's more, to enable the network to restore the different distorted regions


*MM '21, October 20–24, 2021, Virtual Event, China.*

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475608

in the images discriminatively, we also present a discriminative restoration method which can restore the image at different levels according to the distortion degree of the local area of the image. Fig. 1 shows an example of our method.

To more effectively train the network for higher quality image de-distortion, as an important component of our work, we have constructed a large-scale dataset for image de-distortion. To the best of our knowledge, it is the first dataset that is dedicated for evaluating image de-distortion algorithms.

The contributions of this paper are summarized as follows:

- We propose a novel framework for image de-distortion, named Stacked Semantically-Guided Network (SSGN), which stacks multiple semantically guided information to gradually improve the restoration results.
- We present a new discriminative restoration loss function to restore the different distorted regions in the images discriminatively.
- We construct a large-scale dataset for image de-distortion algorithm research and evaluation, which is the first effort that is dedicated to this task.
- We evaluate the proposed approach with extensive qualitative and quantitative experiments. The experimental analyses show the competing results compared with state-of-the-art approaches.

The rest of this paper is organized as follows: Section 2 gives a brief overview of the related works. In Section 3, we introduce our proposed approach for image de-distortion task in detail. Section 4 describes our constructed dataset for image de-distortion and the experimental results, followed by a conclusion in Section 5.

## 2 RELATED WORK

The generative adversarial networks (GANs) [6] have made incredible achievements in multiple image-to-image translation tasks, which include image inpainting, image restoration, and image super-resolution, etc. A lot of GAN-based schemes [1, 3, 10, 11, 17, 18] are proposed by researchers. They are trained with discriminators to determine whether the generated image is true or false, and can generate realistic high-resolution images. At the same time, the discriminators are constantly updated during the training process, with the overall loss being the combination of L1 loss with the mean absolute error and GAN loss using the mean square error.

For image restoration, Deng Xin et al. [4] propose a novel and flexible CNN architecture named Common and Unique information splitting network (CU-Net) to address the general multi-modal image restoration (MIR) and multi-model image fusion (MIF) tasks. As the network architecture is derived from a new proposed multi-modal convolutional sparse coding (MCSC) model, each part of the network becomes interpretable. Yuanbiao Gou et al. [7] design a multi-resolution search space with three task-flexible and interpretable modules. Besides, they propose a novel loss function based on the mean square error (MSE) loss and the binary cross entropy (BCE) loss, which shows a feasible solution to control the model complexity and performance that is highly desirable in the resource-constrained scenarios.

For image inpainting, Lei Zhao et al. [23] propose a conditional image-to-image translation network (UCTGAN) to generate multiple diverse semantically reasonable and visually realistic results for image inpainting. UCTGAN realizes the one-to-one mapping between the instance image space and conditional completion image space, which can significantly improve the diversity of restored images. Zili Yi et al. [21] train the proposed model on small images with the resolution 512x512 and perform the inference on 2K, 4K and 8K high-resolution images, achieving the compelling inpainting quality. They present a novel contextual residual aggregated technique that enables more efficient and high-quality inpainting of ultra-high-resolution images.

To achieve the super resolution of images, Yong Guo et al. [8] propose an invertible flow network, which operates at multiple scale levels. Fuzhi Yang el al. [20] use soft attention and hard attention to find the best matching blocks of the low-resolution (LR) input image in the high-resolution (HR) reference image, and transfer HR textures from the reference to LR image. Their proposed network stacks multiple texture transformers in a cross-scale way with the cross-scale feature integration module (CSFI).

## 3 FRAMEWORK

We propose a stacked semantically-guided learning framework to achieve the discriminative de-distortion of distorted regions in the image. First, we use a deep convolutional neural networks (DCNN) module to extract the semantic features of the distorted image and the original image. Second, the stacked semantically-guided network (SSGN) is introduced to mainly process the distorted image, and outputs the restored image. The overall framework is shown in Fig. 2.

In our proposed framework, we adopt three innovative ideas for the image de-distortion task. First, we use stacked layers from LR to HR to gradually strengthen the network's ability of image de-distortion. Second, we extract rich semantic features and propose the semantically-guided loss (SGL) derivation to equip the network with the capability of rich semantic perception. Third, we propose the discriminative restoration loss (DRL) derivation to discriminatively recover different regions in the image.

### 3.1 Semantic Feature Extraction

Since the deformation operations always deform the local areas of the image, the areas surrounding the person and the edges in the image contain rich potential information that can reveal the local areas where the deformation may occur. The semantic features of the distorted image $x_i$ include the mask $m_i$ of the character, the key-point heatmap $k_i^h$ of the character, the contour heatmap $c_i^h$ of the character, the edge $e_i$ in the image and the discriminative distorted regions of the image. To extract these semantic feature information, we propose a deep convolutional neural networks (DCNN) module composed of three extractors: the heatmap extractor (HE), the edge extractor (EE) and the discriminative region extractor (RE).

HE outputs the mask, consisting of the key-point heatmap and the contour heatmap. For the mask $m_i$, we apply the U square net [14], which is designed for salient object detection (SOD), by loading their pretrained weights for human segmentation. For the key-point heatmap $k_i^h$, we apply the official Openpose network [2] to

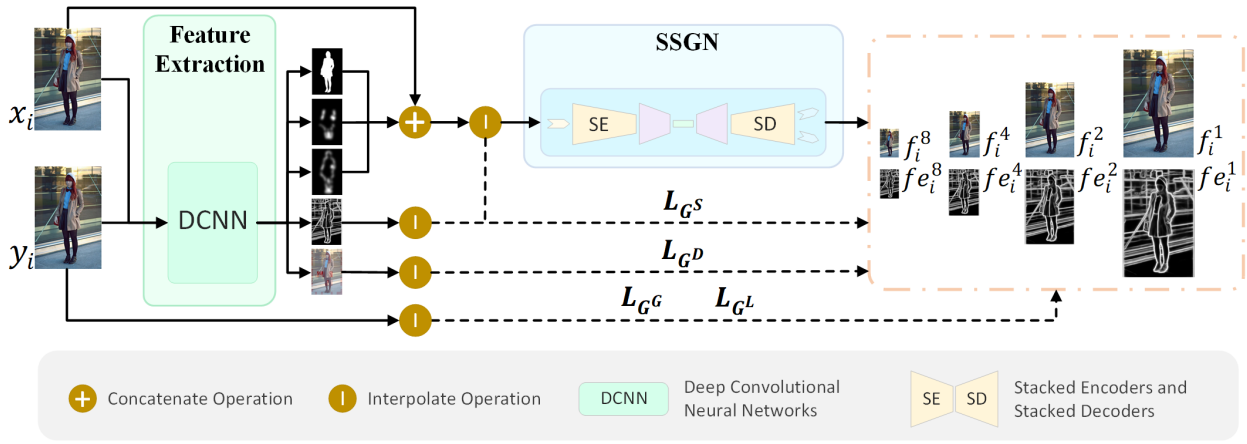


**Figure 2: Stacked semantically-guided learning framework. Given a distorted image $x_i$ and an original image $y_i$, we first use the DCNN to extract adequate semantic features. Then the inputs $x_i^{\hat{s}}$ generated by concatenating $x_i$ with mask, key-point heatmap and contour heatmap, are processed by stacked semantically-guided network (SSGN), whose outputs are $f_i^{\hat{s}}$ and $fe_i^{\hat{s}}$. Finally, the losses $L_{G^G}$, $L_{G^L}$, $L_{G^S}$ and $L_{G^D}$ are calculated and the weight of SSGN is updated by deriving the losses.**

detect the key points of the character in the image and then use the multidimensional gaussian filter to generate the key-point heatmap. For the contour heatmap $c_i^h$, we extract contours from the mask $m_i$ using the opencv library and then use the multidimensional gaussian filter to generate the contour heatmap. With $m_i$, $k_i^h$ and $c_i^h$, we enhance the SSGN ability to recover the surrounding areas of the character.

Since the background distortion mainly occurs on the edges of the image such as railings or walls, we use EE, which contains an accurate edge detector using richer convolutional features (RCF) [12], to extract the edge information $e_i$ of the image.

RE generates the distortion matrix of $x_i$, whose data flow is shown in Fig. 3. First, we divide the distorted image $x_i$ and the original image $y_i$ into 256 regions. Regions from the same position of $x_i$ and $y_i$ are treated as a region pair. The PSNR and SSIM of each region pair are calculated. By setting different thresholds, all regions are divided into different levels according to PSNR or SSIM values to form the distortion matrix. In particular, we use three conditions ($PSNR < 30$ or $SSIM < 0.9$; $PSNR < 25$ or $SSIM < 0.8$; $PSNR < 20$ or $SSIM < 0.7$) to separate all the regions into four levels. The final output of the discriminative region generator $y_{r_i}^{\hat{s}}$ indicates the distortion matrix visualized on a different scale of $y_i$.

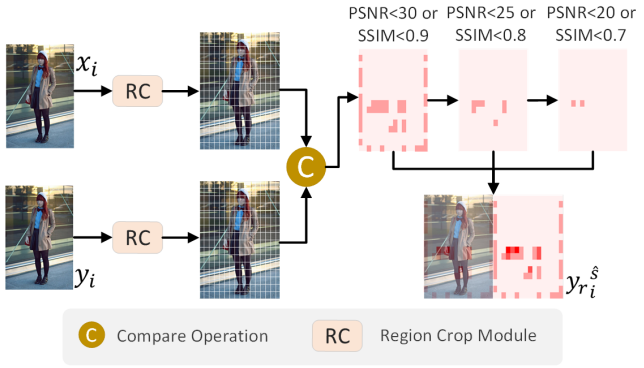


**Figure 3: Data flow of discriminative region extractor. The inputs are $x_i$ and $y_i$, which will be cropped into 256 regions. Then the cropped regions are compared one by one to get the PSNR and SSIM values. We use three conditions ($PSNR < 30$ or $SSIM < 0.9$; $PSNR < 25$ or $SSIM < 0.8$; $PSNR < 20$ or $SSIM < 0.7$) to separate all the regions into four levels.**

## 3.2 Stacked Semantically-Guided Network

It is really difficult to directly generate high-resolution well de-distorted images. To tackle this problem, we propose a stacked semantically-guided network (SSGN) whose graphical depiction is shown in Fig. 2. SSGN gradually improves the recovery effect by stacking from the low-resolution to the high-resolution. At the same time, the network can be trained step by step for each stack, reducing the amount of calculation and the required computing resources during the training process. In addition, we input the semantic information around the body of the character in the image, and force the network to detect the edges of the input image. Finally, SSGN can obtain the ability to perceive the deep semantics of the image, so as to identify the abnormal edge distortion and provide an effective recovery.

As shown in Fig. 2, SSGN could be divided into the following parts: the stacked encoders part, the down-sampling part, the feature processing part, the up-sampling part and the stacked decoders part.

The stacked encoders accept and process multiscale inputs and fuse features of the multiscale inputs. The multiscale inputs first pass through a scale in block (SIB) to unify the number of channels. Then the scale of the higher-level feature is reduced into half by the down-sampling operation, so it can be fused with the feature at the lower level. Until the features of four levels are all fused, the output of the last encoder enters the down-sampling part. The

fusion process of each layer can be formulated as:

$$f\_en_i^{\hat{s}} = Downsample(f\_en_i^{\hat{s}\prime}) + SIB(x_i^{\hat{s}}) \tag{1}$$

where $SIB$ means the scale in block, $f\_en_i^{\hat{s}}$ means the feature of the current layer and $f\_en_i^{\hat{s}\prime}$ means the feature of the higher layer, $x_i^{\hat{s}}$ means the multiscale input of the current layer.

The down-sampling part compresses the size of the feature as much as possible, thereby reducing the weight of the feature processing part. Through the processing of the down-sampling part, the feature size is reduced from 64x64 to 16x16 and the output of the down-sampling part enters the feature processing part.

The feature processing part processes the compressed features and learns the self-attention of deep features by a Dual Attention Network (DANet) [5]. DANet includes two modules: Spatial Attention Module (SAM) and Channel Attention Module (CAM), which extract the spatial attention and channel attention respectively. After passing through the DANet and a convolutional layer, we use five resnet blocks [9] in the innermost layer to process the features. The subsequent processing corresponds to the previous processing, and we also add a dual attention module at the end. The whole process of the feature processing part can be formulated as:

$$\begin{gathered} DA = Conv(CAM(Conv(f\_in_i)) + SAM(Conv(f\_in_i))) \\ f\_out_i = DA(Res(DA(f\_in_i))) \end{gathered} \tag{2}$$

where $DA$ is the formulaic representation of DANet while $CAM$ and $SAM$ are submodules of DANet, $Conv$ means the convolutional operation, $f\_in_i$ and $f\_out_i$ mean the input feature and the output feature.

The up-sampling part deals with the output of the feature processing part and tries to extract and restore the image information from the compressed features. Through the processing of the up-sampling part, the feature size is enlarged from 16x16 to 64x64 for subsequent image restoration.

The stacked decoders part includes an up-sampling operation, a scaled image output block (SOB) and a scaled edge output block (EOB). It starts to restore the image from the expanded feature information coming from the previous part. SOB refines the features pixel by pixel to restore the images with different scales. EOB extracts the edge information from features and performs the edge detection. The process of the outputs could be formulated as:

$$\begin{gathered} f\_de_i^{\hat{s}} = Upsample(Cat(f\_en_i^{\hat{s}\prime}, f\_de_i^{\hat{s}\prime})) \\ f_i^{\hat{s}} = SOB(f\_de_i^{\hat{s}}) \\ fe_i^{\hat{s}} = EOB(f\_de_i^{\hat{s}}) \end{gathered} \tag{3}$$

where $Cat$ means concatenating the feature of the lower layer in the stacked encoders part $f\_en_i^{\hat{s}\prime}$ with the feature of the lower layer in the stacked decoders part $f\_de_i^{\hat{s}\prime}$, $f_i^{\hat{s}}$ and $fe_i^{\hat{s}}$ means the multiscale fake image and the multiscale edge image.

After all the above processing, we unify the multiscale image outputs $f_i^{\hat{s}}$ and multiscale edge outputs $fe_i^{\hat{s}}$ to the maximum scale through the interpolation operation. And finally we choose the largest scale from $f_i^{\hat{s}}$ as the final result $f_i$.

## 3.3 Training Loss

*3.3.1 GAN Loss and L1 Loss.* $GAN$ constraint is carried out through the discriminators. First, each discriminator is trained to identify the authenticity of $y_i$ and $f_i$. Then the generator is trained to deceive the discriminators to identify the input of $f_i$ as $True$, so that the image generated is more similar to a real image. $L1$ constraint is performed by comparing pixel to the pixel error of the original image and the generated fake image. By reducing the $L1$ loss, the difference between the original image and the fake image is also reduced.

According to our validation experiments, we consider the $L1$ constraints with the $l1$-norm and the $GAN$ constraints with the $l2$-norm, which could be expressed as follows:

$$L1(x_i, y_i) = \sum_{i=1}^{n} |y_i - x_i| \tag{4}$$

$$GAN(x_i, y_i) = \sum_{i=1}^{n} (y_i - x_i)^2 \tag{5}$$

where $i$ indicates sample index.

Taking $L1$ and $GAN$ as the basic loss functions, we adopt various loss calculation methods to guide our generator and discriminators. We introduce them from Section 3.3.2 to Section 3.3.4.

*3.3.2 Global Loss and Local Loss.* The global loss calculates the overall difference of images, so that the generator can quickly converge to the original image while the local loss is carried out by randomly dividing each sample image into patches and comparing each patch pair of the original and generated images. We can formulate them as:

$$\begin{gathered} L_{G^G} = \sum_{\hat{s}\in(1,2,4,8)} \left( GAN\left(D^G\left(Cat\left(x_i^{\hat{s}}, f_i^{\hat{s}}\right)\right), p_i^T\right)\right) * \\ \alpha_{GAN} + L1\left(f_i^{\hat{s}}, y_i^{\hat{s}}\right) * \alpha_{L1} \end{gathered} \tag{6}$$

$$\begin{gathered} L_{G^L} = \sum_{\hat{s}\in(1,2,4,8)} \left( \sum_{p=1}^{pn} \left( GAN\left(D^L\left(Cat\left(\mathbb{X}_{pi}^{\hat{s}}, \mathbb{F}_{pi}^{\hat{s}}\right)\right), p_i^T\right)^2 * \right.\right. \\ \left.\left. \alpha_{GAN} + L1\left(\mathbb{F}_{pi}^{\hat{s}}, \mathbb{Y}_{pi}^{\hat{s}}\right) * \alpha_{L1} \right)\right. \end{gathered} \tag{7}$$

where $\hat{s}$ indicates the scale factor and $\hat{s} \in (1, 2, 4, 8)$, $D^G$ and $D^L$ indicate the global discriminator and the local discriminator, $Cat$ indicates the concatenating operation, $x_i^{\hat{s}}$, $y_i^{\hat{s}}$ and $f_i^{\hat{s}}$ indicate the multiscale inputs for SSGN, the multiscale ground truth and the multiscale outputs of SSGN respectively, $p_i^T$ indicates the tensor variable with the value $True$, $p$ indicates the patch index and $pn$ indicates the number of patches cropped from the original image and we set $pn$ = 16 or 32, $\mathbb{X}_{pi}^{\hat{s}}$, $\mathbb{Y}_{pi}^{\hat{s}}$ and $\mathbb{F}_{pi}^{\hat{s}}$ indicate the patches cropped from the multiscale inputs for SSGN, the multiscale ground truth and the multiscale outputs of SSGN respectively. $\alpha_{GAN}$ and $\alpha_{L1}$ are the hyperparameters which control the learning process.

*3.3.3 Semantically-Guided Loss.* We use semantically-guided loss (SGL) to enable SSGN to have the multiple semantic perception of the input image, such as the main human body in the image, the surrounding areas of the human body, the key points areas of the human body and the edges in the image. As for the information of the above areas, we generate it through the DCNN module and input it to SSGN. In the derivation, we overlay the relevant semantic

information on the image to enable the network to focus on these areas. With regard to the edge information, the network senses spontaneously through the loss derivation between an edge and the edge ground truth in the training. SGL could be expressed by:

$$dk = D^K\left(Cat\left(x_i^{\hat{s}} * k_i^{h\hat{s}}, f_i^{\hat{s}} * k_i^{h\hat{s}}\right)\right) \tag{8}$$

$$dc = D^C\left(Cat\left(x_i^{\hat{s}} * c_i^{h\hat{s}}, f_i^{\hat{s}} * c_i^{h\hat{s}}\right)\right) \tag{9}$$

$$\begin{aligned} L_{G^S} = \sum\nolimits_{\hat{s}\in(1,2,4,8)} \Bigg( & \left(GAN\left(dk, p_i^T\right) + GAN\left(dc, p_i^T\right)\right) * \alpha_{GAN} \\ & + L1\left(f_i^{\hat{s}} * k_i^{h\hat{s}}, y_i^{\hat{s}} * k_i^{h\hat{s}}\right) * \alpha_{KP} + L1\left(f_i^{\hat{s}} * c_i^{h\hat{s}}, y_i^{\hat{s}} * c_i^{h\hat{s}}\right) \\ & * \alpha_{CT} + L1\left(fe_i^{\hat{s}}, e_i^{\hat{s}}\right) * \alpha_E \Bigg) \end{aligned} \tag{10}$$

where $D^K$ and $D^C$ indicate the key-point discriminator and the contour discriminator, $k_i^{h\hat{s}}$ and $c_i^{h\hat{s}}$ indicate the multiscale key-point heatmap and the multiscale contour heatmap, $e_i^{\hat{s}}$ and $fe_i^{\hat{s}}$ indicate the multiscale edge ground truth and the multiscale fake edge outputs. $\alpha_{GAN}$, $\alpha_{L1}$, $\alpha_{KP}$, $\alpha_{CT}$ and $\alpha_E$ are the hyperparameters which control the learning process.

*3.3.4 Discriminative Restoration Loss.* We also propose a discriminative restoration loss (DRL) to measure the discriminative regions of the multiscale fake image $f_{r_i}^{\hat{s}}$ and the multiscale fake edge image $fe_{r_i}^{\hat{s}}$ against the multiscale ground truth $y_{r_i}^{\hat{s}}$ and the multiscale edge ground truth $e_{r_i}^{\hat{s}}$, so that SSGN can place different considerations on regions at different levels, to achieve the discriminative image de-distortion. DRL uses the extracted distortion matrix to calculate losses separately for regions at different distortion levels, and sets larger weights to the regions with larger distortion to lead the network to pay more attention to these regions. DRL is defined as:

$$\begin{aligned} L_{G^D} = \sum\nolimits_{\hat{s}\in(1,2,4,8)} \Bigg( & \sum_{r=0}^{3}\left(L1\left(f_{r_i}^{\hat{s}}, y_{r_i}^{\hat{s}}\right) * (r+1)\right) * \\ & \alpha_{DR\_f} + \sum_{r=0}^{3}\left(L1\left(fe_{r_i}^{\hat{s}}, e_{ri}^{\hat{s}}\right) * (r+1)\right) * \alpha_{DR\_e}\Bigg) \end{aligned} \tag{11}$$

where $r$ indicates the region level and $r \in (0, 1, 2, 3)$, $y_{r_i}^{\hat{s}}$,$f_{r_i}^{\hat{s}}$,$e_{r_i}^{\hat{s}}$ and $fe_{r_i}^{\hat{s}}$ indicate the multiscale discriminative regions of the ground truth, the fake outputs, the edge ground truth and the fake edge outputs respectively. $\alpha_{DR\_f}$ and $\alpha_{DR\_e}$ are the hyperparameters which control the learning process.

Finally, we define the loss of generator with all the above functions as:

$$L_G = L_{G^G} + L_{G^L} + L_{G^S} + L_{G^D} \tag{12}$$

## 4 EXPRIMENTS

In this section, we start with the discussion of the dataset for the experiments. Then the training configurations and experimental settings are described. Finally, the effectiveness of our method is shown quantitatively and visually by comparing with the state-of-the-art methods.

### 4.1 Dataset

At present, the task of the image de-distortion has not attracted the attention of researchers, and there is no relevant public dataset

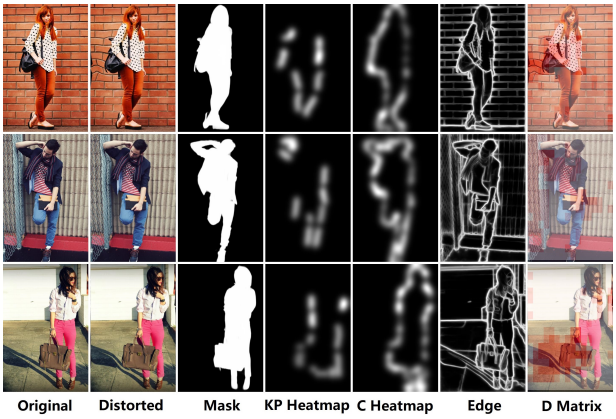


**Figure 4: Samples of self-built dataset. The first two columns are the samples while the others are corresponding extracted semantic features. Images of each column for each line: 1–Original Image, 2–Distorted Image, 3–Mask, 4–Key_point Heatmap, 5–Contour Heatmap, 6–Edge, 7–Distortion Matrix.**

**Table 1: Settings of the hyperparameters**

| Period | $\alpha_{GAN}$ | $\alpha_{L1}$ | $\alpha_{KP}$ | $\alpha_{CT}$ | $\alpha_E$ | $\alpha_{DR\_f}$ | $\alpha_{DR\_e}$ |
|---|---|---|---|---|---|---|---|
| First 3 Epochs | 50 | 1000 | 50 | 50 | 50 | 0.05 | 0.05 |
| Rest Epochs | 50 | 0.05 | 0.05 | 0.05 | 0.05 | 0.02 | 0.02 |

available. Therefore, we create an image distortion dataset that consists of 16080 images. Some samples of our dataset are shown in Fig. 4. Moreover, we have invited some volunteers to make artificial deformations to designated images, so as to obtain distorted images in real-world scenarios.

To create the image distortion dataset, we first crawl more than sixteen thousand images from the Chictopia website, and then use the current state-of-the-art methods to supplement the dataset with the additional information including the character's key points, the character's mask, the character's key-point heatmap, the character's contour heatmap, and the edges in the image to meet the needs of our proposed method. The methods used to generate the above information are introduced in Section 3.1. After that, we design a deforming process based on Moving Least Squares Deformation (MLS) [15] to simulate human's deformation operations. We first cluster contours and key points of the different body parts to generate source points and target points and then deform the local areas of the human body in an original image to obtain the distorted image. Finally the distortion matrix is computed and extracted to guide the network during the training process.

After processing and filtering the raw images, we finally obtain 16080 images as our dataset.

### 4.2 Experimental Details

In our experiment, the training set contains 11474 images and the test set contains 4606 images. We implement our proposed method with Pytorch library and train it on an NVIDIA GeForce RTX 3090 GPU for 30 epochs per stack. Adam optimizer is adopted

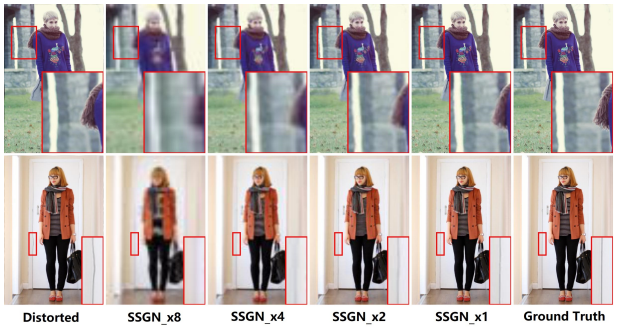


**Figure 5: Ambulation results for stacking SSGN. The distorted areas of the first sample are the skirt and the pillar. The distorted area of the second sample is the door crack. The images from left to right: 1–Distorted Images, 2–SSGN_x8, 3–SSGN_x4, 4–SSGN_x2, 5–SSGN_x1, 6–Ground Truth.**

**Table 2: Quantitative evaluation results without SGL and without DRL. SSGN_xŝ ($\hat{s} \in (1, 2, 4, 8)$) indicates the highest layer among stacked ones. It is performed on 4606 images in the test set.**

| Stack Layer | SSGN_x8 | SSGN_x4 | SSGN_x2 | SSGN_x1 |
|---|---|---|---|---|
| PSNR ↑ | 19.17357 | 20.92960 | 22.21331 | **22.44807** |
| SSIM ↑ | 0.48705 | 0.61905 | 0.73396 | **0.79395** |
| LPIPS ↓ | 0.77061 | 0.62120 | 0.38587 | **0.14839** |
| FID ↓ | 265.35268 | 130.25277 | 48.67030 | **9.33168** |

**Table 3: Quantitative evaluation results with SGL and without DRL. It is performed on 4606 images in the test set.**

| Stack Layer | SSGN_x8 | SSGN_x4 | SSGN_x2 | SSGN_x1 |
|---|---|---|---|---|
| PSNR ↑ | 19.43544 | 21.29471 | 23.44948 | **24.54741** |
| SSIM ↑ | 0.49051 | 0.62565 | 0.74741 | **0.82081** |
| LPIPS ↓ | 0.77140 | 0.62785 | 0.39383 | **0.17410** |
| FID ↓ | 266.49394 | 132.53859 | 51.54481 | **13.28765** |

to optimize SSGN with lr=0.0002. The following are other details in our experiments.

*4.2.1 Settings of the Hyperparameters.* We study the learning performance of the network during the training process under different hyperparameter settings and select more appropriate values shown in Table 1, so that our SSGN can not only consider the clarity and overall texture of the image, but also consider the distortion and recovery of image details.

During the first 3 epochs, we specify the reduction of loss functions using $\alpha_{GAN}$, $\alpha_{L1}$, $\alpha_{KP}$, $\alpha_{CT}$, $\alpha_{E}$ as the *mean* mode and loss functions using $\alpha_{DR\_f}$, $\alpha_{DR\_e}$ as the *sum* mode, aiming to train the network to converge on the whole image.

During the rest of epochs, we specify the reduction of the loss function using $\alpha_{GAN}$ as the *mean* mode and loss functions using $\alpha_{L1}$, $\alpha_{KP}$, $\alpha_{CT}$, $\alpha_{E}$, $\alpha_{DR\_f}$, $\alpha_{DR\_e}$ as the *sum* mode, aiming to avoid the weakening of the local deviation caused by the global mean operation, which allows the network to converge on the local regions in image and recover the distortion more accurately.


**Figure 6: Ambulation results for SGL and DRL derivation. The distorted areas of the first sample are the skirt and the door. The distorted areas of the second sample are the skirt, legs and the signboard. "wo/w" means "without/with". The images from left to right: 1–Distorted Images, 2–SSGN woSGL and woDRL, 3–SSGN wSGL and woDRL, 4–SSGN wSGL and wDRL, 5–Ground Truth.**

**Table 4: Quantitative evaluation results with SGL and with DRL. It is performed on 4606 images in the test set.**

| Stack Layer | SSGN_x8 | SSGN_x4 | SSGN_x2 | SSGN_x1 |
|---|---|---|---|---|
| PSNR ↑ | 20.05408 | 22.31067 | 24.85824 | **27.48422** |
| SSIM ↑ | 0.49785 | 0.64138 | 0.78092 | **0.87983** |
| LPIPS ↓ | 0.76363 | 0.61425 | 0.37629 | **0.13240** |
| FID ↓ | 265.79623 | 129.74353 | 47.40154 | **8.21665** |

**Table 5: Quantitative evaluation results between different methods. In each column, the value with bold red indicates ranking the first place while the value with blue is the second place. It is performed on 4606 images in the test set.**

| Methods | PSNR ↑ | SSIM ↑ | LPIPS ↓ | FID ↓ |
|---|---|---|---|---|
| pix2pix [10] | 22.63999 | 0.75516 | 0.28270 | 28.42515 |
| BicycleGAN [24] | 23.88368 | 0.79708 | 0.24464 | 17.66958 |
| pix2pixHD [18] | 21.88848 | 0.79637 | 0.16365 | 11.38608 |
| HRNet [17] | 21.81392 | 0.76958 | 0.19294 | 12.03030 |
| VQ–VAE [13] | 19.21819 | 0.65698 | 0.39245 | 68.80679 |
| SSGN (Ours) | **27.48422** | **0.87983** | **0.13240** | **8.21665** |

*4.2.2 Stacking Process of SSGN.* SSGN is trained stack by stack for four times, saving the computing resources needed in each training. At first, we train SSGN_x8 layer along with the internal network structures. And then we freeze the trained network weights to train SSGN_x4 layer. The same operations are performed on SSGN_x2 and SSGN_x1 layer, respectively. Each stack is trained for 30 epochs and all the outputs of different layers in SSGN can be observed.

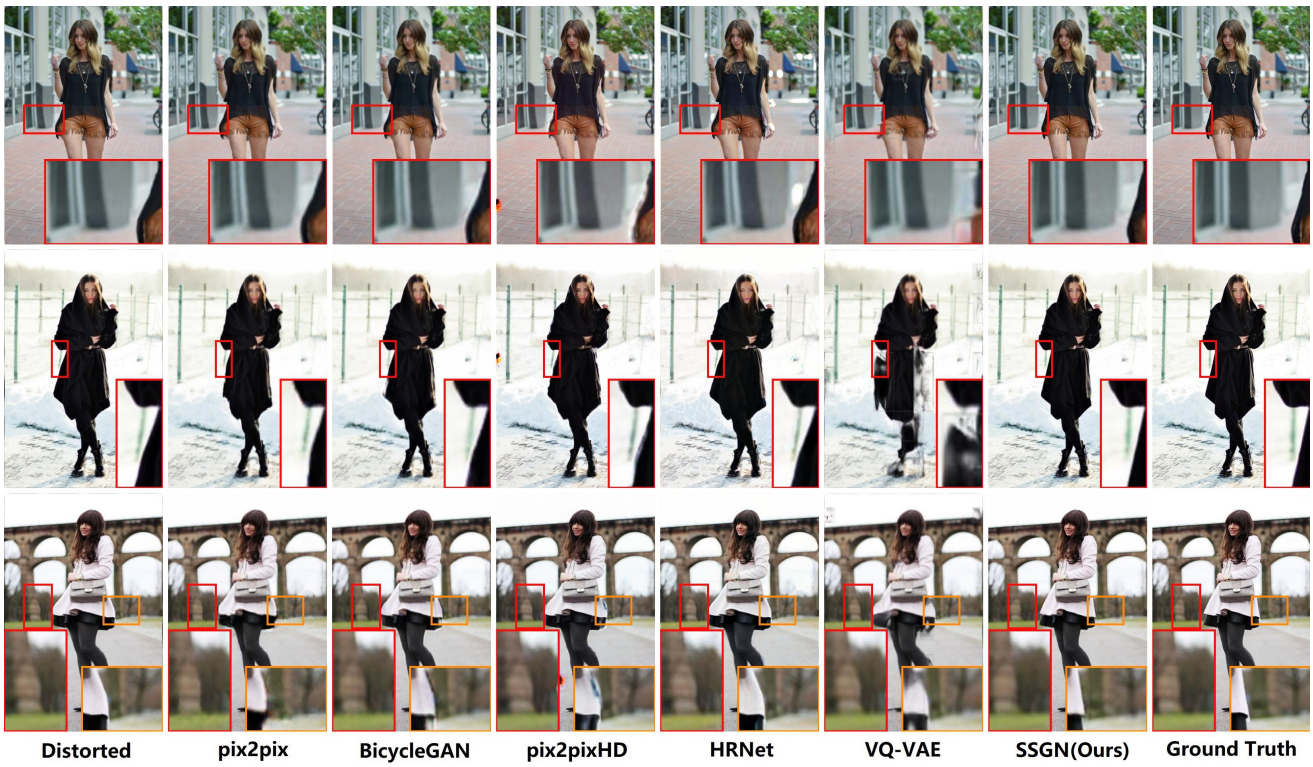


**Figure 7: Visual quality comparison results between different methods on the test images from our dataset. The distorted areas of the images in each row from top to bottom: 1–wall, 2–railing, 3–bridge pillar. The images in each column from left to right: 1–Distorted Images, 2–pix2pix [10], 3–BicycleGAN [24], 4–pix2pixHD [18], 5–HRNet [17], 6–VQ_VAE [13], 7–SSGN(Ours), 8–Ground Truth.**

Note that the output resolution of each layer is gradually 128x128, 256x256, 512x512 and 1024x1024.

*4.2.3 Quality Measures.* The following measures are used to evaluate the performance of different methods: the Peak Signal to Noise Ratio (PSNR), the Structural Similarity Index (SSIM) [19], the Learned Perceptual Image Patch Similarity (LPIPS) [22] and the Fréchet Inception Distance (FID) [16]. The higher PSNR and SSIM values represent the better image quality, while the lower LPIPS and FID values represent the better image quality.

## 4.3 Ablation Study

In this section, we conduct experiments to show the effectiveness of our three innovative ideas for image de-distortion task: (1) four layers stacked from LR to HR; (2) semantically-guiding loss (SGL) derivation; (3) discriminative restoration loss (DRL) derivation. The visual results are shown in Fig. 5 and Fig. 6 and the quantitative results are shown in Table 2, Table 3 and Table 4.

In Fig. 5, we can see that the restoration ability is better when SSGN is stacked higher as the edge and content details of the distorted regions become clearer and more accurate. The same result can be obtained by comparing different columns in each table.

In Fig. 6, if we do not use SGL and DRL, the result looks almost the same as the distorted input and SSGN can't learn any of the de-destortion ability. When we add the SGL derivation, the result shows that SSGN can perceive the correct shape of the skirt, but cannot restore the door or the tree in the background, which indicates the limited ability of the network. Finally, by utilizing SGL and DRL derivation, the network could effectively restore the distortion regions of the input, reaching the level that the naked eye cannot perceive. By comparing the same columns of Table 2, Table 3 and Table 4, we can see the effectiveness of SGL and DRL.

## 4.4 Comparisons with State-of-the-Arts

We adopt several recent state-of-the-art methods with GANs as our baseline models: pix2pix [10], BicycleGAN [24], pix2pixHD [18], HRNet [17] and VQ-VAE [13]. As there are no existing works dedicated to image de-distortion currently, the above approaches that also belong to the field of image generation are the most related to our work.

Because there is no available public dataset in the field of image de-distortion, we use the self-built dataset to fit the relevant codes of these papers and train their networks. To ensure all networks converge as much as possible, we train each network for no less than 30 epochs and measure the qualities of the outputs on 4 indicators mentioned in Section 4.2.3.

As shown in Fig. 7 and Table 5, all of these advanced methods cannot complete the task of high-resolution image de-distortion and our method has achieved the best not only in quantification


**Figure 8: Visual quality comparison results between different methods in the real-world scenarios. The distorted areas of the images in each row from top to bottom: 1–wall, 2–wall, 3–road pile. The images in each column from left to right: 1–Distorted Images, 2–pix2pix [10], 3–BicycleGAN [24], 4–pix2pixHD [18], 5–HRNet [17], 6–VQ_VAE [13], 7–SSGN(Ours).**

but also in visual effect. Pix2pix [10] and BicycleGAN [24] can only slightly recover the distorted regions. Besides, their processing resolution for outputs is much smaller than the others. It can be seen visually that pix2pixHD and HRNet both have abnormal spots in local areas, which shows the difficulty of directly generating the high-resolution well de-distorted images. It confirms the validity of our proposed stacking network structure. It should be pointed out that as VQ-VAE [13] belongs to the field of image inpainting, and it loses much information when processing images, resulting in unsatisfactory results.

### 4.5 Results in Real-world Scenarios

To assess the generalization of the proposed method, we create a dataset of 50 high-resolution distorted images. First, we download images from the Internet. Then we invite several volunteers, all of whom have the experience of editing images, and let them modify the images artificially to obtain a number of distorted images. Finally we evaluate the proposed method and state-of-the-art methods that can be leveraged to alleviate the image distortion. Since the deformed images carry the subjective willingness to correct the images, they cannot be compared with the original ones. Therefore, we show the visual results before and after network restoration. The visual results are shown in Fig. 8 and the proposed method achieves the best visual quality compared with the state-of-the-art methods. We can clearly see that pix2pix [10] tends to exacerbate the object boundary sawtooth. Although BicycleGAN [24] and HRNet [17] restore more smoothly and naturally, they fail to restore the distortion. Besides, both pix2pixHD [18] and VQ-VAE [13] damage the images instead. Only the proposed method can restore real-world distorted images while keeping texture and structure details.

## 5 CONCLUSION

In this paper, we propose a stacked semantically-guided learning network (SSGN) with multiple stacked layers to achieve the discriminative de-distortion of distorted regions in the image, with the aim of effectively dealing with the distortion of images caused by the local deformation operations. During the training process, we utilize semantically-guided loss (SGL) and discriminative restoration loss (DRL) to derive, which guides SSGN to perceive multiple semantics and provide discriminative restoration in regions. As there is no related research before, we construct a large distorted image dataset and use it to train several classical and state-of-the-art networks to demonstrate the effectiveness of our proposed method.

## ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under No. 61872047 and No. 61720106007, the Beijing Nova Program under No. Z201100006820124, the Beijing Natural Science Foundation under No. L191004, the National Key R&D Program of China under No. 2017YFB1003000, and the 111 Project (B18008).

## REFERENCES


[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096* (2018).

[2] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2019. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. *IEEE transactions on pattern analysis and machine intelligence* 43, 1 (2019), 172–186.

[3] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8789–8797.

[4] Xin Deng and Pier Luigi Dragotti. 2020. Deep convolutional neural network for multi-modal image restoration and fusion. *IEEE transactions on pattern analysis and machine intelligence* (2020).

[5] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. 2019. Dual attention network for scene segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3146–3154.

[6] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. *arXiv preprint arXiv:1406.2661* (2014).

[7] Yuanbiao Gou, Boyun Li, Zitao Liu, Songfan Yang, and Xi Peng. 2020. CLEARER: Multi-Scale Neural Architecture Search for Image Restoration. *Advances in Neural Information Processing Systems* 33 (2020).

[8] Yong Guo, Jian Chen, Jingdong Wang, Qi Chen, Jiezhang Cao, Zeshuai Deng, Yanwu Xu, and Mingkui Tan. 2020. Closed-loop matters: Dual regression networks for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5407–5416.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep Residual Learning for Image Recognition. arXiv:1512.03385 [cs.CV]

[10] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.

[11] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4401–4410.

[12] Yun Liu, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. 2017. Richer convolutional features for edge detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3000–3009.

[13] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. 2021. Generating Diverse Structure for Image Inpainting With Hierarchical VQ-VAE. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[14] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. 2020. U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognition* 106 (2020), 107404.

[15] Scott Schaefer, Travis McPhail, and Joe Warren. 2006. Image Deformation Using Moving Least Squares. In *ACM SIGGRAPH 2006 Papers* (Boston, Massachusetts) *(SIGGRAPH '06)*. Association for Computing Machinery, New York, NY, USA, 533–540. https://doi.org/10.1145/1179352.1141920

[16] Maximilian Seitzer. 2020. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid. Version 0.1.1.

[17] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. 2020. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence* (2020).

[18] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 8798–8807.

[19] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[20] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. 2020. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5791–5800.

[21] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. 2020. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7508–7517.

[22] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[23] Lei Zhao, Qihang Mo, Sihuan Lin, Zhizhong Wang, Zhiwen Zuo, Haibo Chen, Wei Xing, and Dongming Lu. 2020. UCTGAN: Diverse Image Inpainting Based on Unsupervised Cross-Space Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5741–5750.

[24] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. 2017. Toward multimodal image-to-image translation. *arXiv preprint arXiv:1711.11586* (2017).