Blind Image Super Resolution with Semantic-Aware Quantized Texture Prior

Chaofeng Chen¹, 4, † Xinyu Shi¹, † Yipeng Qin² Xiaoming Li³ Xiaoguang Han⁴ Tao Yang⁵ Shihui Guo¹

¹School of Informatics, Xiamen University ²School of Computer Science and Informatics, Cardiff University
³Faculty of Computing, Harbin Institute of Technology ⁴SSE, The Chinese University of Hong Kong, Shenzhen ⁵DAMO Academy, Alibaba Group

https://github.com/chaofengc/QuanTexSR

Abstract

A key challenge of blind image super resolution is to recover realistic textures for low-resolution images with unknown degradations. Most recent works completely rely on the generative ability of GANs, which are difficult to train. Other methods resort to high-resolution image references that are usually not available. In this work, we propose a novel framework, denoted as QuanTexSR, to restore realistic textures with the Quantized Texture Priors encoded in Vector Quantized GAN. The QuanTexSR generates textures by aligning the textureless content features to the quantized feature vectors, i.e., a pretrained feature codebook. Specifically, QuanTexSR formulates the texture generation as a feature matching problem between textureless features and a pretrained feature codebook. The final textures are then generated by the quantized features from the codebook. Since features in the codebook have shown the ability to generate natural textures in the pretrain stage, QuanTexSR can generate rich and realistic textures with the pretrained codebook as texture priors. Moreover, we propose a semantic regularization technique that regularizes the pre-training of the codebook using clusters of features extracted from the pretrained VGG19 network. This further improves texture generation with semantic context. Experiments demonstrate that the proposed QuanTexSR can generate competitive or better textures than previous approaches.

1. Introduction

Single image super-resolution (SISR) is a fundamental task in low-level vision, aiming to restore high-resolution (HR) images from their low-resolution (LR) counterparts. Thanks to the incorporation of deep neural networks, previous works [6,7,26,27,33,55] have made significant progress on non-blind SR, which assumes a known degradation process, e.g., bicubic downsampling. However, these methods usually fail in real-world SR tasks where the degradations are unknown, leading to a growing interest in blind SR. Blind SR is intrinsically an ill-posed problem due to the lack of realistic textures in the LR inputs. To restore such missing textures, one line of research resorts to the synthesis power of Generative Adversarial Networks (GANs). For example, the seminal ESRGAN [44] employs GANs to add textures details to the coarse images generated by an SR network pre-trained with L1 loss. Although effective, ESRGAN has two shortcomings: i) it is prone to artifacts due to the notorious unstable GAN training; ii) it struggles to generate different textures for LR patches with similar patterns [43]. Instead of “guessing” the missing textures, another line of research [18,47,58,59] takes advantages of reference images. Their performance is therefore determined by the reference HR images, which can be difficult to locate and are not always available. Addressing this issue, recent works [35,41] turned to implicit texture priors implemented by pretrained GANs. Although bypassing the needs of explicit HR references, these methods are limited to the domain of the pretrained GANs (e.g., face images [3,48]) and cannot generalize to natural images with diverse contents.

In this paper, we propose a novel framework, namely QuanTexSR, for blind SR of natural images. The distinct advantage of our framework is that it addresses the aforementioned limitation of previous works [3,35,41,48] by incorporating quantized texture priors (QTP) that are more suitable for natural image SR. Inspired by the recent VQ-VAE [34,38] and VQGAN [9], we define our QTP as a discrete codebook consisting of a pre-defined number of feature vectors. Such feature vectors contain the information of realistic textures that can be decoded into the target HR images. In this way, we break blind SR into two sub-tasks: i) learning a high-quality QTP; ii) mapping the features of LR inputs to those in QTP for texture restoration. For the first sub-task, we pre-train our QTP with a VQGAN that aims to reconstruct the input HR patches. However, instead of using the vanilla VQGAN, we incorporate semantic information into QTP via a novel semantic regularization dur-
ing pre-training, thereby enhancing the correlation between semantics and textures [43]. For the second sub-task, we first follow ESRGAN [44] and employ a RRDB network pretrained with L1 loss to restore the structure information of the input LR image; then, we feed the output (HR but smooth) images of the RRDB network into the encoder and map their features to their Euclidean nearest neighbour in the pretrained QTP (fixed during training) for SR. To preserve the fidelity of input information, we further propose to add multi-scale U-skip connections to modulate decoder features with encoder features through a novel residual spatial feature transformation (RSFT) block. The encoder and decoder of the pre-trained VQGAN are then finetuned using LR-HR pairs. Extensive experiments demonstrate the effectiveness of our framework. Our contributions can be summarized as follows:

- We propose a novel framework QuanTexSR to restore textures for blind SR using Quantized Texture Priors (QTP) encoded by a pretrained VQGAN network. Compared with GAN-based textures priors, ours enables the generation of more realistic textures with less artifacts for natural images with diverse contents.
- We propose a novel semantic regularization technique for the learning of semantic-aware QTP. Such a regularization enhances the correlation between semantics and texture priors, thereby facilitating the generation of more realistic textures.

2. Related Work

Single Image Super-Resolution (SISR) Starting from the pioneer SRCNN [8], deep neural networks have dominated the design of modern SR algorithms. Since then, various network architectures have been proposed to improve the performance of SISR. For example, Kim et al. [21] proposed a deep version of SRCNN, named VDSR. Thanks to the residual [14] and residual dense blocks [15] that enable training deeper and wider networks, EDSR [27] and RDN [57] were proposed and boosted the performance of SISR. After that, the attention mechanism is also introduced to SISR, such as channel attention [55], spatial attention [4, 33], non-local attention [56], etc. Latest works [6, 26] achieve state-of-the-art performance by employing vision image transformers [28]. These models are trained and evaluated in a non-blind manner, e.g. bicubic downsampling and blurring with known parameters, thereby making it difficult to generalize to SISR with the same degradation type but unseen parameters, let alone those with other degradation types. Addressing this issue, Zhang et al. developed a series of methods [50, 52, 53] for conditional image restoration, where users can control the outputs by changing the conditioned degradation parameters.

Blind SISR Upon the performance saturation of non-blind SISR, recent works turned to the more challenging real-world SISR with unknown degradation (a.k.a. blind SISR). In general, they model complex real-world degradations in either an implicit or an explicit way. Between them, implicit methods [10, 30, 39, 46] aim to learn a degradation network from real-world LR images. In the absence of corresponding ground truth HR images, most of them employed unsupervised image-to-image translation (e.g. Cycle-GAN [61]) while some recent works [40, 49] resort to contrastive learning. On the contrary, explicit methods aim to synthesize “real” LR images by a manually designed degradation process. Specifically, BSRGAN [51] and Real-ESRGAN [42] describe different ways to improve the common image degradation pipeline. Both of them demonstrate much better visual quality than implicit methods in blind SISR. Nevertheless, both implicit and explicit methods rely on the generative power of GANs to generate textures. However, GANs are known to have difficulties in distinguishing some real-world textures from similar degradation patterns, which usually lead to unrealistic textures or over-smoothed regions in the resulting HR images.

Prior-based SISR Since SISR is intrinsically an ill-posed problem, prior-based SISR methods take advantages of extra image priors either explicitly or implicitly. Methods based on explicit prior (a.k.a. RefSR) rely on one or multiple reference HR images which share the same or similar content with the input LR image. To locate the best reference images, various approaches were proposed, including cross-scale correspondences [59], texture transfer [58], transformer network [47], teacher-student [18], internal graph [60], etc. Li et al. [23–25] narrow the image space to faces and achieve impressive performance. Although effective, explicit priors (i.e. HR reference images) are not always available for a given real-world LR image. Therefore, prior-based SISR is more promisingly achieved with a prior distribution (i.e. implicit prior) learnt from a large amount of HR images through GANs or VAEs. Menon et al. [31] first proposed to upscale LR faces by searching the latent space of a pretrained StyleGAN generator [20]. Gu et al. [12] improved it by introducing more latent codes. Pan et al. [36] exploited a BigGAN generator [2] as a prior for versatile image restoration. Although these methods can generate realistic images, they all contain a time-consuming optimization process. Addressing this issue, [3, 41, 48] propose to learn a posterior distribution with a pretrained StyleGAN generator. Specifically, they learn an encoder to project LR images to a latent space shared with the pretrained generator that outputs HR images. Although this approach demonstrates exciting performance for face SR, it hardly works for natural images because learning a GAN for natural images remains a challenging task. In this work, we address the above-mentioned challenge following VQGAN [9] that shows outstanding performance in natural image synthesis.
Figure 1. Texture restoration module $G_e$ of the proposed QuanTexSR, including an encoder $G^t_E$, a decoder $G^t_D$, and the texture quantizer with well-learnt texture prior $f^{tp}$. The feature $f^c$ of the textureless input $\hat{x}$ are quantized into $\hat{f}^c$ with $f^{tp}$ through feature quantization. Since $f^c$ contains rich texture priors, $G^t_D$ can generate $\hat{y}$ with realistic and rich textures.

by learning a quantized texture prior.

3. Methodology

In this section, we will first briefly describe the framework of our proposed QuanTexSR (Sec. 3.1), and then introduce how to implement blind SR with a quantized texture prior (Sec. 3.2), followed by details of our semantically guided prior pretraining (Sec. 3.3). Finally, we will define the objective functions (Sec. 3.4).

3.1. Framework Overview

Given an input LR image $x$ with unknown degradations, we aim to restore the corresponding high-resolution image with realistic textures. Inspired by the two-step training strategy of ESRGAN [44], we employ a two-stage framework that restores the structure and textures sequentially:

- **Stage 1. Structure Restoration Module $G^t$.** This module removes the degradations in $x$ and outputs a “structure” image $\hat{x}$.

  \[ \theta^* = \arg \min \theta \sum ||G^t(x, \theta) - y_i||_1 \]  

  where $G^t$ denotes the RRDB network with parameters $\theta$, and $y_i$ is the corresponding ground truth image. The output image $\hat{x}_i = G^t(x_i, \theta^*)$.

- **Stage 2. Texture Restoration Module $G^f$.** This module adds textures to $\hat{x}$ and generates the final output HR image $\hat{y}$. As shown in Fig. 1, our $G^f$ has three components: an encoder $G^t_E$, a decoder $G^t_D$, and the texture quantizer with well-learnt texture prior $f^{tp} \in \mathbb{R}^{N \times C}$ with $N$ vectors of $C$ dimension. Given an input image $\hat{x}$ obtained by Stage 1, we first feed it into $G^t_E$ and get its feature representation $f^c = G^t_E(\hat{x}) \in \mathbb{R}^{H \times W \times C}$ with $H \times W$ vectors of dimension $C$. Then, we quantize $f^c$ by replacing its $H \times W$ vectors with the closest ones in $f^{tp}$, producing the quantized feature map $\hat{f}^c \in \mathbb{R}^{H \times W \times C}$. Finally, we restore the texture by feeding $\hat{f}^c$ into the decoder $G^t_D$ that is conditioned by the encoder features and get the final output $\hat{y}$.

Furthermore, inspired by [43], we train our $f^{tp}$ with semantic guidance, which benefits QuanTexSR by producing semantically meaningful textures. In the following sections, we will explain the proposed texture restoration process in details.

3.2. Super-resolution with Quantized Texture Prior

As a key component of our texture restoration module $G^f$, our texture quantizer maps encoded feature vectors in $f^c$ to well-learnt ones in our texture prior $f^{tp}$, resulting in quantized features $f^{qc}$ that have been “seen” by the decoder during training that facilitates super-resolution.

**Feature Quantization** Following VQVAE [34], we quantize $f^c$ by locating the Euclidean nearest neighbor of each of its feature vectors in the texture prior $f^{tp}$ as follows:

\[ \hat{f}^c_k = f^{qc}_k, \quad k = \arg \min_j ||f^c_i - f^{qc}_j||_2 \]  

where $f^c, f^{qc} \in \mathbb{R}^{H \times W \times C}$ consists of $H \times W$ feature vectors of dimension $C$, $i \in \{1, 2, \ldots, H \times W\}$, $f^{qc} \in \mathbb{R}^{N \times C}$ consists of $N$ feature vectors of dimension $C$, and $j \in \{1, 2, \ldots, N\}$. After feature quantization, $f^c$ is forced to match the discrete feature distribution defined by $f^{tp}$.

**How can feature quantization help SR?** As we know, image degradation is inherently a one-to-many mapping subject to different types and levels of degradation. From a mathematical point of view, these degradations can be regarded as offsets of high-quality local features in some feature space, where the type and level of degradation correspond to the direction and distance of the offset respectively. Such offsets overlap with each other, thereby making it difficult to find the correct high-quality correspondence of a
degraded feature in the feature space. Heuristically, we address this challenge by mapping a degraded feature to its Euclidean nearest neighbour in a given set of pre-defined high-quality features (i.e., our quantized texture prior). Intuitively, our quantized texture prior partitions the feature space into non-overlapping cells that forms a degradation-based Voronoi diagram. Specifically, we define the $N$ feature vectors $f^E_k$ in our quantized texture prior as the centers of $N$ Voronoi cells. Given an LR feature $f^L$, we compute the Euclidean distance between $f^L$ and all centers $f^tp_k$ to determine which cell $f^E_k$ belongs to\textsuperscript{1}, i.e., which $f^tp_k$ it maps to. In this way, realistic and rich textures can be generated as the decoder inputs are the mapped expressive quantized HR features $f^{tp}_k$ instead of the raw LR features $f^L$.

**Residual Spatial Feature Transformation** Facilitating the generation process of decoder $G_D$, we propose Residual Spatial Feature Transformation (RSFT), which extends the Spatial Feature Transform (SFT) [41, 43] by adding extra shortcut connections (Fig. 1) that fuse content and texture features at decoder layers. Specifically, our RSFT modulates decoder features with affine transformation parameters ($\alpha, \beta$) generated by the corresponding encoder features and is formulated as:

$$\alpha, \beta = \Phi_l(F^E_l)$$

$$F^D_l = (1 + \alpha) \cdot F^D_l + \beta$$

where $\Phi_l$ is made up of a few convolution layers, $F^E_l$ denotes the content feature output of encoder layer $l$, and $F^D_l$ denotes the texture feature output of the corresponding decoder layer. $F^E_l$ and $F^D_l$ are connected by the U-Skip connections of our RSFT block. In addition, since $G_D$ is initialized with a pretrained VQGAN, the identity shortcut makes full usage of the original feature flow, and allow the network to learn the residual only in texture restoration stage.

\textsuperscript{1}In some rare cases (of zero probability), $f^E_k$ has the same nearest distances to multiple $f^L$, i.e. on the boundary of the Voronoi cells. In these cases, we randomly map $f^L$ to one of the centers.

### 3.3. Semantic-Guided QTP Pretraining

As above-mentioned, our quantized texture prior (QTP) is obtained from a pretraining stage that will be detailed as follows. To emphasize the correspondence, we use the same notations as in previous sections. The backbone of our pretraining is a VQVAE that aims to learn the reconstruction $y'$ of a given HR image $y$ using the decoder $G_D$ and the features of $y$ that are encoded and quantized by $G_E$ and a codebook $f^{tp}$ respectively:

$$y' = G_D(f^y) \approx y$$

where $f^y$ is the quantized version of $f^u$ using Eq. (2). $f^y$ is the feature of $y$ encoded by $G_E$. Since the feature quantization operation is non-differentiable, we follow [9, 34] and simply copy the gradients from $G_D$ to $G_E$ for backpropagation. The model and codebook can be trained with the following objective function:

$$\mathcal{L}_{VQ}(G_E, G_D, f^{tp}) = \|y' - y\|_1 + \|sg[G_E(y)] - f^y\|^2_2 + \beta\|sg[f^y] - G_E(y)\|^2_2$$

where $sg[\cdot]$ is the stop-gradient operation, $\beta = 0.25$ [9, 34].

**Semantic Guidance** As indicated by the vanilla setting in Eq. (6), the codebook (i.e. our quantized texture prior) $f^{tp}$ is learnt purely by gradient descent where similar patterns are clustered independent of their semantics. To learn a semantic-aware $f^{tp}$, we regularize its training with another semantic-aware codebook $f^{VGG}$ that is obtained by applying mini-batch K-means to features of HR training images that are extracted by a pretrained VGG19 network. Then, we can obtain the quantized feature of $f^y$ with $f^{VGG}$ using Eq. (2), denoted as $f^y-VGG$. We use $f^y-VGG$ as a semantic regularization of $f^{tp}$ and extends Eq. (6) as

$$\mathcal{L}_{VQ-VGG} = \mathcal{L}_{VQ} + \gamma\|f^y - f^y-VGG\|^2_2$$

where $\gamma$ is a weighting factor empirically set to 0.01. Note that we follow [9] and also use perceptual loss and adversarial loss in the pretraining. Please see Fig. 2 for an intuitive illustration of our method.

In summary, our semantic-guided QTP pretraining encourages the texture generation to be conditioned on semantics, thereby enabling the generation of more realistic and natural textures. We will verify this in our experiments.

### 3.4. Training Objectives

**Reconstruction Loss** We follow [9, 41] and employ L1 and perceptual losses as our reconstruction loss, formulated as

$$\mathcal{L}_{rec} = \lambda_{L1}\|\hat{y} - y\|_1 + \lambda_{per}\|\phi(\hat{y}) - \phi(y)\|^2_2$$

where $\phi$ is a pretrained VGG-16 network. $\lambda_{L1}$ and $\lambda_{per}$ are weights of the L1 and perceptual losses respectively.
Feature Quantization Loss Since the quantization process is non-differentiable, we use the same trick as in the pre-training stage that copies gradients from the decoder $G_E^t$ to the encoder $G_E^t$ directly. Besides, we also keep the last term of Eq. (6), a.k.a. “commitment loss”, to update $G_E^t$ that

$$L_{quant} = \beta ||sg[f^e] - G_E^t(\hat{x})||_2^2 \quad (9)$$

Adversarial Loss Although texture priors already contain rich texture information, we still need adversarial loss to help us find better matching features in the quantization process. We follow [42] and adopt a U-Net discriminator $D$ with spectral normalization [32]. Similar to [5], we use a hinge loss and define the generator loss as

$$L_{adv} = \lambda_{adv} \sum_i -\mathbb{E}[D(y_i)] \quad (10)$$

For simplicity, the discriminator loss is omitted here.

Overall Loss The overall loss is defined as

$$L_{total} = L_{rec} + L_{quant} + L_{adv} \quad (11)$$

where the weights for each loss are set as: $\lambda_{L1} = \lambda_{per} = 1$, $\beta = 0.25$, $\lambda_{adv} = 0.1$.

4. Implementation Details

4.1. Datasets and Evaluation Metrics

Training Dataset We follow BSRGAN [51] and build a training set that includes DIV2K [1], Flickr2K [27], DIV8K [13] and 10,000 face images from FFHQ [19]. We use the following ways to generate the training patches: (1) crop non-overlapping $512 \times 512$ patches; (2) filter patches with few textures; (3) for well-aligned faces in FFHQ, we perform random resize with scale factors between $[0.5, 1.0]$ before cropping to avoid content bias. The final training dataset contains 136,205 HR patches of size $512 \times 512$. We use the same degradation model as BSRGAN to generate corresponding LR images.

Synthetic Testing Dataset To ensure a fair comparison, we use a mixed degradation model of two recent works BSRGAN and Real-ESRGAN, denoted as bsganplus, to generate LR testsets for DIV2K validation set and 5 classical benchmarks, i.e., Set5, Set14, BSD100, Urban100 and Manga109. The diversity of test images guarantees a comprehensive evaluation of model performance.

Real-world Testing Dataset We test our model on three recent real-world benchmarks, including RealSR [41], DRealSR [45] and DPED-iphone [16]. We test models with an upscale factor of 4 for these real-world datasets. Images from RealSR and DRealSR are captured by DSLR cameras, and contain 100 and 93 images respectively. DPED-iphone includes 100 LR images captured by smartphone cameras. The LR images in DPED-iphone are usually more corrupted than those from RealSR and DRealSR.

Evaluation Metrics For synthetic test datasets with ground truth images, we employ the well-known perceptual metric, LPIPS [54] score, to evaluate the perceptual quality of generated images. We also report the results of the widely used PSNR, SSIM scores for references. For real-world benchmarks, there are usually no ground truth images, therefore we adopt the well-known no reference metric NIQE score for quantitative comparison.

4.2. Training Details

In both the QTP pretraining and SR training, we use an Adam [22] optimizer with $\beta_1 = 0.9, \beta_2 = 0.99$. The learning rates for both the generator and discriminator are fixed as $0.0001$ throughout the training. We use our pretrained QTP networks to initialize the encoder and decoder weights before the training of texture stage. During texture restoration stage, the codebook is fixed while the encoder and decoder are finetuned. Both our QTP and SR networks are trained for 400k iterations with a batch size of 16, and the HR image size is fixed as $256 \times 256$ for both $\times 2$ and $\times 4$ upscale factors. We implemented our model with PyTorch [37] and train them on 2 GeForce RTX 3090 GPUs.

5. Experiments

5.1. Analysis of Texture Priors

In this experiment, we visualize the features in the codebook $f^{tp}$ (i.e., our texture prior), which facilitates the understanding of the proposed framework by answering two questions: i) what texture priors are encoded in $f^{tp}$? ii) how are they correlated to the semantics?

As shown in Fig. 3, we visualize the texture priors encoded in $f^{tp}$ by projecting features to RGB pixel space with pretrained decoder $G_D^t$ where $1 \times 1$ features are mapped to $16 \times 16$ RGB patches. Specifically, we explore how textures are encoded by single codes and combinations of different codes:

- Fig. 3(a) shows that individual codes alone can represent some basic texture elements. However, when the same code is tiled onto a bigger feature map, e.g., $4 \times 4$, the decoder tends to preserve the color while producing a smooth image. This implies that a single code is not enough to represent complex textures.
- Fig. 3(b) shows that complex and realistic textures can be generated by combining a number of different code samples, which indicate that the pretrained $f^{tp}$ indeed learns to encode rich texture priors. In addition, it can be observed that different combinations of code samples correspond to different semantics, such as, (1)
Table 1. Quantitative comparison with state-of-the-art methods on synthetic benchmarks for blind SR. LR images are generated with a mixed degradation model of BSRGAN [51] and Real-ESRGAN [42]. PSNR/SSIM ↑: the higher, the better; LPIPS ↓: the lower, the better. LPIPS scores can better reflect texture quality, and the best and second performance are marked in red and blue.

| Method       | Scale | DIV2K Valid | Set5 | Set14 | BSD100 | Urban100 | Manga109 |
|--------------|-------|-------------|------|-------|--------|----------|----------|
|              |       | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| CDC          | ×2    | 24.93 | 0.6588 | 0.5153 | 22.74 | 0.5347 | 0.6229 | 23.64 | 0.5282 | 0.7073 | 20.94 | 0.5118 | 0.7001 | 21.60 | 0.6345 | 0.5723 |
| DAN          | ×2    | 24.69 | 0.5729 | 0.6588 | 22.79 | 0.5083 | 0.5639 | 23.46 | 0.4923 | 0.6384 | 20.93 | 0.4793 | 0.6603 | 21.78 | 0.5832 | 0.5639 |
| DASR(W)      | ×2    | 24.74 | 0.5767 | 0.6304 | 22.81 | 0.5110 | 0.5720 | 23.49 | 0.4958 | 0.6508 | 20.94 | 0.4819 | 0.6606 | 21.80 | 0.5878 | 0.5587 |
| BSRGAN       | ×2    | 26.06 | 0.7075 | 0.3182 | 26.57 | 0.7799 | 0.2027 | 24.59 | 0.6475 | 0.3013 | 22.76 | 0.6391 | 0.3199 | 24.64 | 0.7678 | 0.2285 |
| Real-ESRGAN+ | ×2    | 25.50 | 0.6963 | 0.2993 | 26.73 | 0.7771 | 0.2157 | 23.65 | 0.6299 | 0.3023 | 21.66 | 0.6148 | 0.2876 | 23.88 | 0.7698 | 0.2135 |
| SwinIR-GAN   | ×2    | 25.53 | 0.6868 | 0.3313 | 27.07 | 0.7993 | 0.2093 | 23.76 | 0.6364 | 0.3128 | 21.54 | 0.6195 | 0.3003 | 23.36 | 0.7705 | 0.2283 |
| Ours         | ×2    | 25.56 | 0.6783 | 0.2876 | 26.53 | 0.7333 | 0.1879 | 23.73 | 0.6026 | 0.2970 | 22.26 | 0.6155 | 0.2918 | 24.23 | 0.7641 | 0.2002 |

| Method       | Scale | DIV2K Valid | Set5 | Set14 | BSD100 | Urban100 | Manga109 |
|--------------|-------|-------------|------|-------|--------|----------|----------|
|              |       | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| CDC          | ×4    | 23.11 | 0.5850 | 0.7132 | 19.99 | 0.5077 | 0.7168 | 20.38 | 0.4551 | 0.7377 | 21.75 | 0.4800 | 0.7707 | 19.42 | 0.4568 | 0.7345 |
| DAN          | ×4    | 24.22 | 0.5929 | 0.6881 | 20.85 | 0.5319 | 0.6771 | 21.44 | 0.4937 | 0.6758 | 22.52 | 0.4818 | 0.7438 | 20.20 | 0.4757 | 0.7228 |
| DASR(W)      | ×4    | 24.19 | 0.5920 | 0.7021 | 20.87 | 0.5336 | 0.6972 | 21.43 | 0.4953 | 0.6950 | 22.49 | 0.4818 | 0.7576 | 20.18 | 0.4752 | 0.7400 |
| BSRGAN       | ×4    | 24.91 | 0.6500 | 0.3596 | 21.63 | 0.5573 | 0.4683 | 22.17 | 0.5165 | 0.4173 | 22.95 | 0.5042 | 0.3405 | 20.91 | 0.5386 | 0.3874 |
| Real-ESRGAN+ | ×4    | 23.80 | 0.6414 | 0.3696 | 21.31 | 0.5449 | 0.5068 | 21.54 | 0.5288 | 0.4271 | 22.43 | 0.5035 | 0.4693 | 19.90 | 0.5282 | 0.3838 |
| SwinIR-GAN   | ×4    | 24.13 | 0.6479 | 0.3543 | 20.91 | 0.5128 | 0.5115 | 21.58 | 0.5041 | 0.4487 | 22.23 | 0.4925 | 0.4447 | 20.01 | 0.5300 | 0.3592 |
| Ours         | ×4    | 23.72 | 0.6256 | 0.3450 | 20.80 | 0.5046 | 0.4633 | 21.59 | 0.4906 | 0.3984 | 22.03 | 0.4838 | 0.4448 | 20.46 | 0.5337 | 0.3899 |

Based on the above discussion, we conjecture that the individual codes in $f^{tp}$ represent simple texture elements, while the diverse semantics are encoded in the combinations of multiple codes.

5.2. Comparison with Existing Methods

We compare the proposed QuanTexSR with several state-of-the-art methods for blind SR, including CDC [45], DAN [29], DASR(W) [40], RealSR [17], BSRGAN [51], Real-ESRGAN+ [42] and SwinIR-GAN [26]. Specifically, CDC proposed a divide-and-conquer architecture; DAN, DASR(W) and RealSR learned degradation models from LR inputs; BSRGAN, Real-ESRGAN+ and SwinIR-GAN used synthetic training data generated by handcrafted degradation models. We use the original codes and weights from the official public github repositories for all competing methods. Quantitative and qualitative results on both synthetic and real benchmarks are shown in Table 1.

(a) Textures generated with tiled single code. The tiled feature size are: $1 \times 1$, $2 \times 2$, $3 \times 3$, $4 \times 4$ (from top to bottom) (b) Textures generated with random combination with different number of codes. The size of combined feature map is $16 \times 16$.

Figure 3. Visualization of texture priors encoded with pretrained codebook $f^{tp}$. Semantic textures emerge when different codes are combined together, such as 1 grass, 2 plant and 3 water.
Figure 4. Visual comparisons on two examples from synthesize benchmarks with upscale factor of 2 (first row) and 4 (second row). Thanks to the texture prior, our model is able to restore realistic and faithful textures even when the inputs are severely corrupted. As for the competitive works, some have difficulties to remove degradation, *i.e.*, DAN and DASR(W), and the others generate artifacts or extra unrealistic textures, *i.e.*, BSRGAN, Real-ESRGAN+, SwinIR-GAN.

Figure 5. Visual comparisons on two real-world examples with upscale factor 4. Our model shows better performance on natural texture restoration such as leaves (first row) and twigs (second row).
Table 2. Quantitative comparison with state-of-the-art methods on real-world benchmarks. NIQE ↓: the lower, the better. The best and second performance are marked in red and blue.

| Datasets          | Bicubic | DAN     | RealSR  | CDC     | DASR(W) | BSRGAN    | Real-ESRGAN+ | SwinIR-GAN | Ours      |
|-------------------|---------|---------|---------|---------|---------|-----------|--------------|------------|-----------|
| RealSR [41]       | 6.2438  | 6.5673  | 6.8041  | 6.2376  | 5.7335  | 4.7832    | 4.7644       | 5.0176     |           |
| DRealSR [45]      | 6.5766  | 7.0720  | 7.7213  | 6.6359  | 9.1446  | 6.1362    | 4.8458       | 4.7053     | 4.4956    |
| DPED-iphone [16]  | 6.0121  | 6.1414  | 5.5855  | 6.2738  | 6.9887  | 5.9906    | 5.2631       | 4.9468     | 4.8774    |

**Comparison on Synthetic Benchmarks** As Tab. 1 shows, our QuanTexSR outperforms competing methods in LPIPS scores on most benchmarks (4 out of 6). Note that we focus on the LPIPS scores as it better captures the perceptual quality than other metrics (e.g., PSNR/SSIM) [41, 42, 48, 51, 54]. In addition, it can be observed that: in general, methods that learn the degradations, such as DAN and DASR(W), perform much worse than those using manually designed degradation models, which indicates the difficulties in learning complex real-world degradations. Furthermore, we compare the SR results qualitatively through visual inspection in Fig. 4. It can be observed that in both scenarios, BSRGAN, Real-ESRGAN and SwinIR-GAN cannot remove the noise in the LR images and resulting in unrealistic HR images with artefacts. We conjecture that such artefacts might originate from the unstable adversarial training of GANs. In contrast, thanks to the quantized texture priors, our method does not have such problems and generates higher quality results.

**Comparison on Real-world Benchmarks** To make a fair comparison, we compare our method against state-of-the-art ones on three large real-world benchmarks and evaluate the results using a standard no-reference IQA metric NIQE. As Tab. 2 show, our method outperforms competing methods in 2 out of 3 real-world benchmarks, which clearly demonstrates the effectiveness of our framework. In Fig. 5, it can be observed that our QuanTexSR produces sharp and clear textures, while the other methods fail to generate realistic textures but produce blurry results.

**5.3. Ablation Study**

We conduct ablation experiments on two variations of our framework with the same number of parameters: i) a model without feature quantization and ii) a model with feature quantization but without semantic guidance. As Tab. 3 shows, removing either feature quantization or semantic guidance worsens the performance. Furthermore, through visual inspection on the generated images (Fig. 6), we conclude that the proposed feature quantization helps to recover the grass textures, while the semantic guidance further improves the texture quality and make it more realistic.

![Figure 6. Ablation studies on feature quantization and semantic guidance. Please zoom in for best view.](image)

Table 3. LPIPS scores of different model variations on synthetic benchmark with upscale factor of 4.

| Models       | w/o quant | w/o semantic | Full model |
|--------------|-----------|--------------|------------|
| Div2k valid  | 0.3621    | 0.3568       | **0.3450** |

![Figure 7. Failure case: a building image with straight lines.](image)

**5.4. Limitations**

From 2 out of the 6 synthetic benchmarks (especially Urban100) where our method was not the best (Tab. 1), we discovered a limitation of our QuanTexSR: it favors natural textures over artificial textures, e.g., the straight lines that dominate the building images in Urban100. Our method usually generates curved lines instead (Fig. 7). A similar phenomenon also occurs in neural texture synthesis [11]. We leave the solution of this problem to future work.

**6. Conclusion**

In this paper, we investigated the usage of quantized texture prior for realistic texture restoration for blind SR. In particular, we propose an encoder-decoder like framework which incorporates pretrained quantized texture priors (QTP) through feature quantization. We integrate se-
semantic information to QTP with features from pretrained VGG19 network. We have analyzed how textures and semantics are encoded in QTP through feature visualization. Quantitative and qualitative experiments on both synthetic and real-world benchmarks demonstrate the superiority of the proposed QuanTexSR in natural texture recovery.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In CVPRW, July 2017. 5
[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In ICLR, 2019. 2
[3] Kelvin CK Chan, Xintao Wang, Xiangyu Xu, Jinwei Gu, and Chen Change Loy. Glean: Generative latent bank for large-factor image super-resolution. In CVPR, pages 14245–14254, 2021. 1, 2
[4] Chaofeng Chen, Dihong Gong, Hao Wang, Zhifeng Li, and Kwan-Yee K. Wong. Learning spatial attention for face super-resolution. In IEEE TIP, 2020. 2
[5] Chaofeng Chen, Xiaoming Li, Yang Lingbo, Xianhui Lin, Lei Zhang, and KKY Wong. Progressive semantic-aware style transformation for blind face restoration. In CVPR, 2021. 5
[6] Hanxing Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In CVPR, 2021. 1, 2
[7] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In CVPR, pages 11065–11074, 2019. 1
[8] Chao Dong, Chen Change Loy, Kaiming He, and Xiaou Tang. Learning a deep convolutional network for image super-resolution. In ECCV, pages 184–199. Springer, 2014. 2
[9] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In CVPR, pages 12873–12883, 2021. 1, 2, 4
[10] Manuel Fritsche, Shuhang Gu, and Radu Timofte. Frequency separation for real-world super-resolution. In ICCVW, pages 3599–3608, 2019. 8
[11] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. NeurIPS, 28:262–270, 2015. 8
[12] Jinjin Gu, Yujun Shen, and Bolei Zhou. Image processing using multi-code gan prior. In CVPR, pages 3012–3021, 2020. 2
[13] Shuhang Gu, Andreas Lugmayr, Martin Danelljan, Manuel Fritsche, Julien Lamour, and Radu Timofte. Div8k: Diverse 8k resolution image dataset. In ICCVW, pages 3512–3516. IEEE, 2019. 5
[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, pages 770–778, 2016. 2
[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In CVPR, pages 4700–4708, 2017. 2
[16] Andrey Ignatov, Nikolay Kobychev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool. Dsr-quality photos on mobile devices with deep convolutional networks. In ICCV, pages 3277–3285, 2017. 5, 8
[17] Xiaozhong Ji, Yun Cao, Ying Tai, Chengjie Wang, Jilin Li, and Feiyue Huang. Real-world super-resolution via kernel estimation and noise injection. In CVPRW, pages 466–467, 2020. 6
[18] Yuming Jiang, Kelvin CK Chan, Xintao Wang, Chen Change Loy, and Ziwei Liu. Robust reference-based super-resolution via c2-matching. In CVPR, pages 2103–2112, 2021. 1, 2
[19] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. ICLR, 2018. 5
[20] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In CVPR, pages 8110–8119, 2020. 2
[21] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In CVPR, pages 1646–1654, 2016. 2
[22] Dieterik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014. 5
[23] Xiaoming Li, Chaofeng Chen, Shangchen Zhou, Xianhui Lin, Wangmeng Zuo, and Lei Zhang. Blind face restoration via deep multi-scale component dictionaries. In ECCV, pages 399–415. Springer, 2020. 2
[24] Xiaoming Li, Wenyu Li, Dongwei Ren, Hongzhi Zhang, Meng Wang, and Wangmeng Zuo. Enhanced blind face restoration with multi-exemplar images and adaptive spatial feature fusion. In CVPR, pages 2706–2715, 2020. 2
[25] Xiaoming Li, Ming Liu, Yuting Ye, Wangmeng Zuo, Liang Lin, and Ruigang Yang. Learning warped guidance for blind face restoration. In ECCV, pages 272–289, 2018. 2
[26] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In ICCVW, 2021. 1, 2, 6
[27] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Cynthia Rudin. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In CVPR, pages 2437–2445, 2020. 2
[32] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 5

[33] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Hai Feng Shen. Single image super-resolution via a holistic attention network. In *ECCV*, pages 191–207. Springer, 2020. 1, 2

[34] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *NeurIPS*, 2017. 1, 3, 4

[35] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. In *ECCV*, 2020. 1

[36] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. Exploiting deep generative prior for versatile image restoration and manipulation. In *ECCV*, pages 262–277. Springer, 2020. 2

[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, volume 32, pages 8026–8037, 2019. 5

[38] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *NeurIPS*, pages 14866–14876, 2019. 1

[39] Ziyu Wan, Bo Zhang, Dongdong Chen, Pan Zhang, Dong Chen, Jing Liao, and Fang Wen. Bringing old photos back to life. In *CVPR*, pages 2747–2757, 2020. 2

[40] Longguang Wang, Yingqian Wang, Xiaoyu Dong, Qingyu Xu, Junchang Yang, Wei An, and Yulun Guo. Unsupervised degradation representation learning for blind super-resolution. In *CVPR*, pages 10581–10590, 2021. 2, 6

[41] Xintao Wang, Yu Li, Honglin Zhang, and Ying Shan. Towards real-world blind face restoration with generative facial prior. In *CVPR*, pages 9168–9178, 2021. 1, 2, 4, 5, 8

[42] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. *ICCVW*, 2021. 2, 5, 6, 8

[43] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*, June 2018. 1, 2, 3, 4

[44] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCVW*, pages 0–0, 2018. 1, 2, 3

[45] Pengxu Wei, Ziwei Xie, Hannan Lu, Zongyuan Zhan, Qixiang Ye, Wangmeng Zuo, and Liang Lin. Component divide-and-conquer for real-world image super-resolution. In *ECCV*, pages 101–117. Springer, 2020. 5, 6, 8

[46] Yunxuan Wei, Shuhang Gu, Yawei Li, Radu Timofte, Longcun Jin, and Hengjie Song. Unsupervised real-world image super resolution via domain-distance aware training. In *CVPR*, pages 13385–13394, 2021. 2

[47] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtuo Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *CVPR*, June 2020. 1, 2

[48] Tao Yang, Peiran Ren, Xuansong Xie, and Lei Zhang. Gan prior embedded network for blind face restoration in the wild. In *CVPR*, pages 672–681, 2021. 1, 2, 8

[49] Jiashui Zhang, Shijian Li, Fangneng Zhan, and Yingchen Yu. Blind image super-resolution via contrastive representation learning. *arXiv preprint arXiv:2107.00708*, 2021. 2

[50] Kai Zhang, Luc Van Gool, and Radu Timofte. Deep unfolding network for image super-resolution. In *CVPR*, pages 3217–3226, 2020. 2

[51] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. *ICCV*, 2021. 2, 5, 6, 8

[52] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *CVPR*, pages 3262–3271, 2018. 2

[53] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Deep plug-and-play super-resolution for arbitrary blur kernels. In *CVPR*, pages 1671–1681, 2019. 2

[54] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 5, 8

[55] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *ECCV*, pages 286–301, 2018. 1, 2

[56] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. Residual non-local attention networks for image restoration. In *ICLR*, 2019. 2

[57] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *ICLR*, pages 2472–2481, 2018. 2

[58] Zhifei Zhang, Zhaowen Wang, Zhe Lin, and Hairong Qi. Image super-resolution by neural texture transfer. In *CVPR*, pages 7982–7991, 2019. 1, 2

[59] Haitian Zheng, Mengqi Ji, Haoqian Wang, Yebin Liu, and Lu Fang. Learning texture transformer network for image super-resolution via domain-distance aware training. In *CVPR*, pages 101–117. Springer, 2020. 1, 2

[60] Haijun Zhou, Jiawei Zhang, Wangmeng Zuo, and Liang Lin. Component-wise network for image super-resolution. In *ICCV*, pages 2747–2757, 2018. 2

[61] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017. 2