Real-World Blind Super-Resolution via Feature Matching with Implicit High-Resolution Priors

Chaoeng Chen∗  
chaofenghust@gmail.com  
School of Informatics, Xiamen University  
Xiamen, China

Xinyu Shi∗  
xinyu.shi@uwaterloo.ca  
University of Waterloo  
Waterloo, Canada

Yipeng Qin  
QinY16@cardiff.ac.uk  
Cardiff University  
Cardiff, United Kingdom

Xiaoming Li  
csxmli@gmail.com  
Harbin Institute of Technology  
Harbin, China

Xiaoguang Han  
hanxiaoguang@cuhk.edu.cn  
SSE, The Chinese University of Hong Kong  
Shenzhen, China

Tao Yang  
yangtao9009@gmail.com  
DAMO Academy, Alibaba Group  
Hangzhou, China

Shihui Guo†  
guoshihui@xmu.edu.cn  
School of Informatics, Xiamen University  
Xiamen, China

Figure 1: Comparison between our FeMaSR and two latest works, Real-ESRGAN+ [44] and SwinIR-GAN [27] on a low resolution image with complex blind degradations. Our method can recover realistic hairs for the squirrel thanks to the implicit high-resolution priors. Please zoom in for the best view.

ABSTRACT

A key challenge of real-world image super-resolution (SR) is to recover the missing details in low-resolution (LR) images with complex unknown degradations (e.g., downsampling, noise and compression). Most previous works restore such missing details in the image space. To cope with the high diversity of natural images, they either rely on the unstable GANs that are difficult to train and prone to artifacts, or resort to explicit references from high-resolution (HR) images that are usually unavailable. In this work, we propose Feature Matching SR (FeMaSR), which restores realistic HR images in a much more compact feature space. Unlike image-space methods, our FeMaSR restores HR images by matching distorted LR image features to their distortion-free HR counterparts in our pretrained HR priors, and decoding the matched features to obtain realistic HR images. Specifically, our HR priors contain a discrete feature codebook and its associated decoder, which are pretrained on HR images with a Vector Quantized Generative Adversarial Network (VQ-GAN). Notably, we incorporate a novel semantic regularization in VQGAN to improve the quality of reconstructed images. For the feature matching, we first extract LR features with an LR encoder consisting of several Swin Transformer blocks and then follow a simple nearest neighbour strategy to match them with the pretrained codebook. In particular, we equip the LR encoder with residual shortcut connections to the decoder, which is critical to the optimization of feature matching loss and also helps to complement the possible feature matching errors. Experimental results show that our approach produces more realistic HR images than previous methods. Codes are released at https://github.com/chaofengc/FeMaSR.

CCS CONCEPTS

- Computing methodologies → Computational photography.
KEYWORDS
Blind Super-Resolution; FeMaSR; Feature Matching; High-Resolution Prior; VQGAN

ACM Reference Format:
Chaofeng Chen, Xinyu Shi, Yipeng Qin, Xiaoming Li, Xiaoguang Han, Tao Yang, and Shihui Guo. 2022. Real-World Blind Super-Resolution via Feature Matching with Implicit High-Resolution Priors. In Proceedings of the 30th ACM International Conference on Multimedia (MM ’22), October 10–14, 2022, Lisboa, Portugal. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3503161.3547833

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. Due to the incorporation of deep neural networks, previous works [6, 7, 27, 28, 34, 57] 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, i.e., blind SR.

Blind SR is intrinsically an ill-posed problem because the complex and unknown distortions in the LR inputs have disrupted many details. Some works [12, 40, 53, 62] exploited assumptions of the classical degradation model to explicitly estimate the blur kernel and noise. As a result, most of them can only handle several simplified cases of the classical degradation model, and are a far cry from real-world SR solutions. Other works [44, 47, 52, 56] resort to the synthesis power of Generative Adversarial Networks (GANs) to generate the missing textures. Although effective, these approaches are prone to artifacts due to the notorious unstable GAN training. Instead of “guessing” the missing textures, another line of research [19, 48, 60, 61] takes advantages of reference images. Their performance is therefore determined by the reference HR images, which are not always available. Addressing this issue, recent works [36, 43] turned to implicit high-resolution 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, 49]) and cannot generalize to natural images with diverse contents¹.

In this paper, we propose a novel SR framework based on feature matching, namely FeMaSR, for blind SR of real-world images. The distinct advantage of our framework is that it addresses the aforementioned limitations of previous works by matching LR features to a set of HR features in the pretrained implicit HR priors (HRP). Inspired by the recent VQ-VAE [35, 39] and VQGAN [9], we define our HRP as the combination of a discrete codebook consisting of a pre-defined number of feature vectors and the corresponding pretrained decoder. The 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 HRP; ii) mapping the features of LR inputs to the codebook in HRP for distortion removal and detail recovery. For the first sub-task, we pre-train our HRP with a VQGAN that aims to reconstruct the input HR patches. However, instead of using the vanilla VQGAN, we incorporate semantic information into HRP via L2 regularization with perceptual features from VGG19, thereby enhancing the correlation between semantics and codebook features. For the second sub-task, we follow SwinIR [27] and utilize several swin transformer blocks to encode the LR inputs. The LR encoder is then trained with losses between LR features and ground truth HR features selected from the pretrained codebook. Especially, we found that the feature matching loss is difficult to optimize with fixed HRP. To solve this problem, we introduce multi-scale residual shortcut connections from LR feature space to decoder features. These residual connections enable direct gradient flow from pretrained decoder to the LR encoder, thus making it easier to optimize the LR encoder. Besides, it also helps to complement the possible feature matching errors. Since HRP contains rich semantic-aware HR information of natural images, the proposed FeMaSR is able to recover higher quality textures, see Fig. 1. Our contributions can be summarized as follows:

- We propose a novel framework FeMaSR for blind SR using HRP encoded by a pretrained VQGAN network. Compared with previous works, the FeMaSR formulates SR as a feature matching problem between LR features and distortion-free HR priors, and therefore enables the generation of more realistic images with less artifacts for real-world SR.
- We introduce semantic regularization for the pretrain of semantic-aware HRP. Such a regularization enhances the correlation between semantics and HRP, thereby facilitating the generation of more realistic textures.
- We design an LR encoder with residual shortcut connections to the HRP for feature matching. The proposed framework can better match the LR features with distortion free HR features, and also complement the matching errors.

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. [22] proposed a deep version of SRCNN, named VDSR. Thanks to the residual [15] and residual dense blocks [16] that enable training deeper and wider networks, EDSR [28] and RDN [59] were proposed and boosted the performance of SISR. After that, the attention mechanism is also introduced to SISR, such as channel attention [57], spatial attention [4, 34], non-local attention [58], etc. Latest works [6, 27] achieve state-of-the-art performance by employing vision image transformers [29]. 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 [51, 53, 54] 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.

¹To our knowledge, all state-of-the-art GANs that can synthesize high-quality and high-resolution images are dedicated to a specific domain (e.g., StyleGAN [21]).
Figure 2: Framework of the proposed FeMaSR. It contains two stages: pretrain of high-resolution prior, and super-resolution via feature matching. We first pretrain a VQGAN to learn an implicit representation of high-resolution patches, i.e., the codebook $\mathcal{Z}$ and decoder $G$. Then the LR encoder $E_l$ is optimized to find the best matching features of the LR inputs $x$ in the codebook $\mathcal{Z}$. Since $\mathcal{Z}$ and $G$ are pretrained to reconstruct high resolution patches, FeMaSR is able to generate clearer results with less artifacts.

Between them, implicit methods [10, 31, 41, 42, 47] 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 [64]) while some recent works [50] resort to contrastive learning. On the contrary, explicit methods aim to synthesize “real” LR images by a manually designed degradation process. Specifically, BSRGAN[52] and Real-ESRGAN[44] 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 [61], texture transfer [60], transformer network [48], teacher-student [19], internal graph [63], etc. Li et al. [24–26] 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. [32] first proposed to upscale LR faces by searching the latent space of a pretrained StyleGAN generator [21]. Gu et al. [13] improved it by introducing more latent codes. Pan et al. [37] 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, 43, 49] 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 and can be regarded as high-quality priors for image synthesis.

3 METHODOLOGY

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. As shown in Fig. 2, we employ a two-stage framework to pretrain the High-Resolution Priors (HRP) and conduct feature matching sequentially:

- **Stage I, Pretraining of High-Resolution Priors.** We use HR patches to pretrain a VQGAN [9] consisting of an encoder $E$, a discrete codebook $\mathcal{Z}$, and a decoder $G$. Inspired by [45], we train the VQGAN with semantic guidance that enhances the correlation of textures and semantics. We call the codebook $\mathcal{Z}$ and decoder $G$ HRP. After pretraining, our HRP approximately encodes the complete information of HR images and allows the reconstruction of them by feeding their corresponding feature codes $z \in \mathcal{Z}$ to $G$. 



Stage II, Super-Resolution via Feature Matching. Given the HRP (i.e., \(Z\) and \(G\)) obtained in Stage I, we argue that blind SR is equivalent to a feature matching problem that aims to match the feature codes of LR inputs \(\hat{z}\) to those of their HR counterparts \(z \in Z\). By feeding \(G\) with the correctly matched HR feature codes \(z\), we can obtain the clean and realistic HR images required in blind SR. To address the optimization challenges posed by the quantization process of VQGAN, we further propose the incorporation of a residual shortcut module to the LR encoder. This not only facilitates training but also complements the feature matching errors, which further boosts the quality of the resulting HR images.

Details are described in the following sections.

3.2 Pretraining of High-Resolution Priors

We first give a brief review of VQGAN. As illustrated in Fig. 2, the input LR image \(y \in \mathbb{R}^{H \times W \times 3}\) is first passed through the encoder \(E\) to produce its output feature \(\hat{z} = E(y) \in \mathbb{R}^{H \times W \times n_z}\), where \(n_z\) is the feature dimension. Then the discrete representation of \(\hat{z}\) is calculated by finding the nearest neighbours of each element \(\hat{z}_i \in \mathbb{R}^{n_z}\), in the codebook \(Z \in \mathbb{R}^{K \times n_z}\) as follows:

\[
z_i = Z_k, \quad k = \arg\min_j \|\hat{z}_i - Z_j\|_2, \quad (1)
\]

where \(z \in \mathbb{R}^{H \times W \times n_z}\), \(K\) is the codebook size, \(i \in \{1, 2, \ldots, H \times W\}\), and \(j \in \{1, 2, \ldots, K\}\). After that, \(y\) is reconstructed by \(z\) with the decoder \(G\):

\[y' = G(z) \approx y. \quad (2)\]

Since the feature quantization operation of Eq. (1) is non-differentiable, we follow [9, 35] and simply copy the gradients from \(G\) to \(E\) for backpropagation. Therefore, the model and codebook can be trained end-to-end with the following objective function:

\[
\mathcal{L}_VQ(E, G, Z) = ||y' - y||_1 + ||\text{sg}(z) - z||_2^2 + \beta||\text{sg}(z) - \hat{z}||_2^2, \quad (3)
\]

where \(\text{sg}[]\) is the step-gradient operation, and \(\beta = 0.25\) according to [9, 35]. With the pretrained VQGAN, any high resolution images \(y\) from the training set can be reconstructed with their corresponding feature vectors in \(Z\) and the decoder \(G\). We therefore call them HRP in this work.

Semantic Guidance. As indicated by the vanilla setting in Eq. (3), the codebook \(Z\) is learned purely by gradient descent where similar patterns are clustered independent of their semantics. Meanwhile, Wang et al. [45] pointed out that semantic guidance leads to better texture restoration. This motivates us to incorporate semantic information in the pretraining of VQGAN. To be specific, we regularize the training of codebook \(Z\) with perceptual features from a pretrained VGG19 network by adding a regularization term \(L_r\) to \(L_{VQ}\) and have

\[
\mathcal{L}'_{VQ} = \mathcal{L}_{VQ} + L_r = \mathcal{L}_{VQ} + \gamma \|\text{CONV}(z) - \phi(y)\|_2^2 \quad (4)
\]

where \(\phi(y)\) denotes a simple convolution layer to match the dimension of \(z\) and \(\phi(y)\), \(\phi\) denotes the pretrained VGG19, and \(\gamma\) is a weighting factor empirically set to 0.1. Note that we follow [9] and also use perceptual loss and adversarial loss in the pretraining.

In summary, our semantic-guided HRP pretraining encourages the texture restoration to be conditioned on semantics, thereby enabling the restoration of more realistic and natural textures.

3.3 Super-Resolution via Feature Matching

With the pretrained HRP, i.e., \(Z\) and \(G\), the SR task is turned into a feature matching problem between LR inputs \(x\) and \(Z\). Denote the LR encoder as \(E_l\), the problem can be formulated as

\[
\arg\min_{\theta} \mathcal{L}(G(q[E_l(x, \theta), Z]), y), \quad (5)
\]

where \(\theta\) is the learnable parameter of \(E_l\), \(q[\cdot]\) denotes the feature matching process same as Eq. (1), and \(\mathcal{L}\) denotes the loss functions (which will be described in the following section). We first make a brief discussion about why we want to transform the SR task to a feature matching process and how can it help:

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., the pretrained codebook \(Z\)). Intuitively, the codebook with discrete features partitions the feature space into non-overlapping cells that form a degradation-based Voronoi diagram. As demonstrated in Fig. 2, we define the \(K\) feature vectors \(z_k\) in \(Z\) as the centers of \(K\) Voronoi cells. Given an LR feature \(z_l\), we compute the Euclidean distance between \(z_l\) and all centers \(z_k\) to determine which cell \(z_l\) belongs to, i.e., which \(z_k\) it maps to. In this way, realistic and rich textures can be generated as the decoder inputs are mapped to expressive HR features \(z_k\) instead of the raw LR features \(z_l\).

Despite the advantages of feature matching, the optimization of Eq. (5) is quite challenging because of the complex LR inputs. For this purpose, we introduce a powerful LR encoder \(E_l\) consisting of two parts: feature extraction module and residual shortcut module.

Feature Extraction. As shown in Fig. 2, the design of the feature extraction module basically follows SwinIR [27]. It is composed of a shallow feature extraction head and a deep feature extraction block. The deep feature extraction block applies the same stack of a shallow feature encoder block and a deep feature extraction block. The deep feature extraction block applies the same stack of residual swin transformer layers as SwinIR, while the shallow feature extraction block applies the same stack of a shallow feature encoder block. The deep feature extraction block applies the same stack of residual swin transformer layers as SwinIR, while the shallow feature extraction block applies the same stack of a shallow feature encoder block.

In this work, we have \(S_{up} = S_{down} \times 8\) as the decoder \(G\) upscale the \(z \in \mathbb{R}^{H \times W}\) by \(\ast 8\). Denote the feature extraction module as \(H_F\), we have:

\[
z_l = H_F(x), \quad (6)
\]

\(\ast 8\)In some rare cases (of zero probability), \(z_l\) has the same nearest distances to multiple \(z_k\), i.e., on the boundary of the Voronoi cells. In these cases, we randomly map \(z_l\) to one of the centers.
Residual Shortcut Module To better utilize the HRP, we further introduce multi-scale residual connections between $\tilde{z}^l$ and the decoder $G$, as shown in Fig. 2. To be specific, we use several upsampling blocks $H_\text{up}$ to upscale LR features $\tilde{z}^l$ and add them as residuals to the decoder $G$, i.e.,

$$ f_0 = z, \hat{f}_0 = \tilde{z}^l $$ (7)

$$ \hat{f}_i = G_\text{up}(f_{i-1}) + H_\text{up}(\hat{f}_{i-1}), i \in \{1, 2, 3, \ldots\} $$ (8)

where $G_\text{up}$ and $H_\text{up}$ are the $i$-th upsampling blocks in $G$ and $H_F$ respectively, and $\hat{f}_{i-1}$ are the input features to them.

Our residual shortcut module has two main benefits. First, it sidesteps the non-differentiable quantization process in VQGAN, which greatly eases the optimization difficulty. Second, we observed that these extra residual connections have also learned to complement the potential errors in feature matching and can further boost the performance of blind SR.

3.4 Training Objectives

The gradients to update $E_i$ come from three parts: feature matching losses, image reconstruction losses, and adversarial loss.

**Feature Matching Loss** This loss is dedicated to the training of $E_i$. We first obtain the ground truth latent representation of $y$, i.e., $z_{gt} = q(E(y), \mathcal{Z})$, and then calculate the L2 loss and the Gram matrix loss for LR features

$$ L_{fema} = \beta \|z - z_{gt}\|_2^2 + \alpha \|\psi(z^l - \psi(z_{gt}))\|_2^2, $$ (9)

where $\psi$ calculates the Gram matrix of features, and $\alpha$ is its weight. The Gram matrix loss, also called style loss, has been shown to be helpful to restore textures [11].

**Reconstruction Loss** We follow [9, 43] and employ L1 and perceptual losses as our reconstruction loss, formulated as

$$ L_{rec} = \lambda_{L1}\|\tilde{y} - y\|_1 + \lambda_{per}\|\phi(\tilde{y}) - \phi(y)\|_2^2 $$ (10)

where $\phi$ is a pretrained VGG-16 network, $\lambda_{L1}$ and $\lambda_{per}$ are weights of the L1 and perceptual losses respectively.

**Adversarial Loss** Although our HRP already contains rich texture information, we still need an adversarial loss to help us find better-matching features in the feature matching process. We follow [44] and adopt a U-Net discriminator $D$ with spectral normalization [33]. Similar to [5], we use a hinge loss and define the generator loss as

$$ L_{adv} = \lambda_{adv} \sum_i -\mathbb{E}[D(\tilde{y}_i)] $$ (11)

For simplicity, the discriminator loss is omitted here.

**Overall Loss** The overall loss is defined as

$$ L_{total} = L_{fema} + L_{rec} + L_{adv} $$ (12)

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

4 IMPLEMENTATION DETAILS

4.1 Datasets and Evaluation Metrics

**Training Dataset** We follow BSRGAN [52] and build a training set that includes DIV2K [1], Flickr2K [28], DIV8K [14] and 10,000 face images from FFHQ [20]. 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. More details are provided in the supplementary material. The final training dataset contains 136,205 HR patches of size $512 \times 512$. We use the same degradation model as BSRGAN\(^3\) 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 bsrGAN_plus\(^3\), 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 [43], DRealSR [46] and DPEDE-iphone [17]. 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. DPEDE-iphone includes 100 LR images captured by smartphone cameras. The LR images in DPEDE-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 [55] 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 HRP pretraining and SR training, we use an Adam [23] 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. During feature matching stage, the codebook $\mathcal{Z}$ and decoder $G$ are fixed. Both our HRP and SR networks are trained 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 [38]. The model pretraining stage takes about 3 days on 2 GeForce RTX 3090 GPUs and the SR stage takes about 4 days on the same device.

5 EXPERIMENTS

5.1 Visualization of HRP

In this experiment, we visualize the features in the codebook $\mathcal{Z}$ with pretrained $G$, which facilitates the understanding of the proposed framework by answering two questions: i) what priors are encoded in HRP ii) how are they correlated to the semantics?

As shown in Fig. 3, we visualize the priors encoded in $\mathcal{Z}$ by projecting features to RGB pixel space with pretrained decoder $G$. In other words, we obtain the RGB patches of each vector $z_j \in \mathcal{Z}$ with $G(z_j)$, where the size of RGB patches is $8 \times 8$. Specifically, we explore how textures are encoded by single codes and combinations of different codes:

\(^3\)https://github.com/cszn/BSRGAN
Table 1: Quantitative comparison with state-of-the-art methods on synthetic benchmarks. LR images are generated with a mixed degradation model of BSRGAN [52] and Real-ESRGAN [44]. 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 | PSNR | SSIM | LPIPS | PSNR | SSIM | LPIPS |
| CDC          | x2    | 24.93 | 0.6293 | 0.6588 | 25.35 | 0.6747 | 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          | x2    | 24.69 | 0.5729 | 0.6219 | 25.27 | 0.6278 | 0.4658 | 22.79 | 0.5083 | 0.5639 | 23.46 | 0.4925 | 0.6384 | 20.93 | 0.4793 | 0.6605 | 21.78 | 0.5832 | 0.5639 |
| DASR(W)      | x2    | 24.74 | 0.5767 | 0.6304 | 25.31 | 0.6312 | 0.4735 | 22.81 | 0.5110 | 0.5720 | 23.49 | 0.4958 | 0.6508 | 20.94 | 0.4819 | 0.6696 | 21.80 | 0.5878 | 0.5587 |
| BSRGAN       | x2    | 26.60 | 0.7073 | 0.3182 | 27.65 | 0.7799 | 0.2027 | 24.59 | 0.6475 | 0.3013 | 24.88 | 0.5967 | 0.3769 | 22.76 | 0.6391 | 0.3199 | 24.64 | 0.7678 | 0.2285 |
| Real-ESRGAN+ | x2    | 25.50 | 0.6963 | 0.2993 | 26.73 | 0.7771 | 0.2157 | 23.65 | 0.6299 | 0.3023 | 24.11 | 0.5860 | 0.3433 | 21.66 | 0.6148 | 0.2876 | 23.88 | 0.7696 | 0.2135 |
| SwinIR-GAN   | x2    | 25.33 | 0.6868 | 0.3313 | 27.07 | 0.7793 | 0.2093 | 23.76 | 0.6364 | 0.3128 | 23.83 | 0.5717 | 0.3707 | 21.54 | 0.6195 | 0.3003 | 23.56 | 0.7705 | 0.2283 |
| Ours         | x2    | 25.26 | 0.6680 | 0.2753 | 26.46 | 0.7470 | 0.1964 | 23.38 | 0.5982 | 0.2852 | 23.83 | 0.5599 | 0.3264 | 21.90 | 0.5956 | 0.2777 | 23.64 | 0.7407 | 0.2192 |

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. Some numbers of competitive methods are taken from [44].

| Datasets     | Bicubic | DAN       | RealSR   | CDC | DASR(W) | BSRGAN | Real-ESRGAN+ | SwinIR-GAN | Ours  |
|--------------|---------|-----------|----------|-----|---------|--------|--------------|------------|-------|
| RealSR [43]  | 6.2436  | 6.5673    | 6.8041   | 6.2376 | 8.1918  | 5.7355 | 4.7832       | 4.7644     | 4.7434 |
| DRealSR [46] | 6.5766  | 7.0720    | 7.7213   | 6.6359 | 9.1446  | 6.1362 | 4.8458       | 4.7053     | 4.1987 |
| DPED-iphone [17] | 6.0121  | 6.1414    | 5.5855   | 6.2738 | 6.9887  | 5.9906 | 5.2631       | 4.9468     | 5.1066 |

- Fig. 3(b) shows that complex and realistic textures can be generated by combining several different code samples, which indicates that the pretrained Z 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) grass, (2) plant and (3) water. Please see the supplementary materials for more examples.

Based on the above discussion, we conjecture that the individual codes in Z 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 [46], DAN [30], DASR(W) [42], RealSR [18], BSRGAN [52], Real-ESRGAN+ [44] and SwinIR-GAN [27]. 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-world benchmarks are reported as follows.

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

Figure 3: Visualization of texture priors encoded with pretrained codebook Z. Semantic textures emerge when different codes are combined, such as (1) grass, (2) plant and (3) water.

- 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 × 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.
Figure 4: Visual comparisons on two examples from synthesize benchmarks with upscale factor of 2 (first column) and 4 (second column). Thanks to the HRP, 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 tend to be oversmooth, i.e., BSRGAN, Real-ESRGAN+, SwinIRGAN. Please zoom in for best view.

Figure 5: Visual comparisons on two real-world example with upscale factor 4. Our model can remove degradations and generate feasible details at the same time, while other GAN based methods tend to be either over-textured (first row) or over-smooth (second row). Please zoom in for best view.

Comparison on Synthetic Benchmarks As Tab. 1 shows, our QuanTexSR outperforms competing methods in LPIPS scores on most benchmarks (5 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) \[43, 44, 49, 52, 55\]. 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 the first column, BSRGAN, Real-ESRGAN and SwinIR-GAN mistake the feather textures as noises and remove them. And in the second column, although the distortions are removed successfully, they all fail to generate feasible textures for the trees. In contrast, thanks to the semantic-aware HRP, 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 shows, 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 FeMaSR produces sharp and clear textures without generating artifacts, while the other methods either fail to remove degradations or tend to be over-textured and over-smooth. Please see the supplementary materials for more results.

5.3 Ablation Study

We conduct ablation experiments on four variations of our framework as shown in Tab. 3 to validate our design: Model-A, a baseline network by discarding Stage I, feature matching and residual shortcuts. It has a similar architecture with SwinIR, and is trained with GAN from scratch; Model-B, Model-A with pretrained decoder; Model-C, Model-B with pretrained codebook and feature matching; FeMaSR, full model with HRP and residual shortcuts; Model-D, FeMaSR based on HRP without semantic guidance.
Effectiveness of Residual Shortcut

As claimed in Sec. 3, residual shortcut helps optimization of feature matching process and complements possible matching errors. We verify them by removing the residual shortcut in training (Model-C) and testing stage respectively. As we can see in Fig. 6(a), the feature matching loss $L_{fema}$ decreases much faster with residual shortcut. This indicates that residual shortcut is essential for the optimization of $L_{fema}$.

We can also observe a clear performance drop of model C without residual shortcut in Tab. 3 and Fig. 7. We further demonstrate how residual shortcut helps to complement feature matching errors in Fig. 6(b). We can notice that model with disabled residual shortcut can already remove the distortions to a large extent. The residual shortcut mainly complements the color and edges.

Effectiveness of HRP

Model-[A, B and FeMaSR] validate the necessities of $Z$ and $G$ in HRP. As discussed above, the performance drop of Model-C is mainly due to the optimization difficulty brought by feature matching. Therefore, we do not use it to validate HRP. It can be observed that Model-B is better than Model-A since the pretrained decoder helps to stabilize GAN training. However, both Model-A and Model-B cannot handle complex distortions without feature matching and tend to generate artifacts, see Fig. 7. Meanwhile, the full model, FeMaSR, can make full use of HRP in both $G$ and $Z$, and thereby has the best performance.

Effectiveness of Semantic Guidance

We provide reconstruction training loss curve and LPIPS score in Stage I to show the benefits of semantic guidance. It can be seen that VQGAN with semantic guidance converges faster and performs better, resulting in a better HRP. This finally helps to improve the restoration performance, see Tab. 3 and Fig. 7.

6 CONCLUSION

In this paper, we have investigated the usage of implicit high-resolution priors (HRP) encoded in the codebook and associated decoder of a pretrained VQGAN for real-world blind SR. In particular, we formulate the SR task to a feature matching problem between the LR features and distortion free HR feature codebook. Because HRP is distortion free and fixed during SR stage, our FeMaSR is able to generate more realistic results with less artifacts than previous GAN based approaches. To train a better HRP, we integrate semantic information to HRP with features from pretrained VGG19 network. To facilitate optimization of feature matching loss, we introduce multi-scale residual shortcut connections to the pretrained decoder. Quantitative and qualitative experiments on both synthetic and real-world benchmarks demonstrate the superiority of the proposed FeMaSR for real-world LR images.

7 ACKNOWLEDGEMENT

This work was done at GAP Lab, CUHK SZ which is directed by Prof. Xiaoguang Han, and is supported by Alibaba Innovative Research, National Natural Science Foundation of China (62072383, 61702433, 62077039), the Fundamental Research Funds for the Central Universities (20720210044, 20720190006).
[63] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, and Chen Change Loy. 2020. Cross-Scale Internal Graph Neural Network for Image Super-Resolution. In NeurIPS.

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