Blind Image Super-resolution with Rich Texture-Aware Codebooks

Rui Qin
School of Software, Tsinghua University & BNRist, China
qr20@mails.tsinghua.edu.cn

Ming Sun
Kuaishou Technology, China
sunming03@kuaishou.com

Fangyuan Zhang
School of Software, Tsinghua University & BNRist, China
zhangfy19@mails.tsinghua.edu.cn

Xing Wen
Kuaishou Technology, China
td.wenxing@gmail.com

Bin Wang*
School of Software, Tsinghua University & BNRist, China
wangbins@tsinghua.edu.cn

ABSTRACT

Blind super-resolution (BSR) methods based on high-resolution (HR) reconstruction codebooks have achieved promising results in recent years. However, we find that a codebook based on HR reconstruction may not effectively capture the complex correlations between low-resolution (LR) and HR images. In detail, multiple HR images may produce similar LR versions due to complex blind degradations, causing the HR-dependent only codebooks having limited texture diversity when faced with confusing LR inputs. To alleviate this problem, we propose the Rich Texture-aware Codebook-based Network (RTCNet), which consists of the Degradation-robust Texture Prior Module (DTPM) and the Patch-aware Texture Prior Module (PTPM). DTPM effectively mines the cross-resolution correlation of textures between LR and HR images by exploiting the cross-resolution correspondence of textures. PTPM uses patch-wise semantic pre-training to correct the misperception of texture similarity in the high-level semantic regularization. By taking advantage of this, RTCNet effectively gets rid of the misalignment of confusing textures between HR and LR in the BSR scenarios. Experiments show that RTCNet outperforms state-of-the-art methods on various benchmarks by up to ***0.16*** ~ ***0.46dB***.

CCS CONCEPTS

• **Computing methodologies** → **Image processing**.

KEYWORDS

Neural networks, blind super-resolution, codebook, texture

ACM Reference Format:
Rui Qin, Ming Sun, Fangyuan Zhang, Xing Wen, Bin Wang. 2023. Blind Image Super-resolution with Rich Texture-Aware Codebooks. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29-November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3581783.3611917

*Corresponding author. This work was supported by the National Natural Science Foundation of China under Grant 62072271.

MM '23, October 29-November 3, 2023, Ottawa, ON, Canada

ACM ISBN 979-8-4007-0108-5/23/10.
https://doi.org/10.1145/3581783.3611917

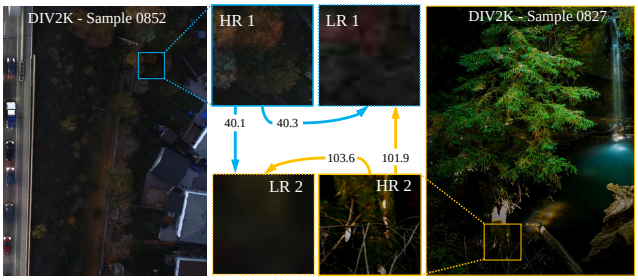


Figure 1: Confusing LR samples with different HR textures processed with the same random degradation used in [7, 8, 66] (including various noise, blur, and compression). The MSE in RGB space on the line indicates the patch distance.

1 INTRODUCTION

Blind Super-Resolution (BSR) aims to realistically reconstruct high-resolution (HR) images from low-resolution (LR) images with unknown degradation [7, 8, 21, 30, 34, 55, 57]. To avoid the General Adversarial Network (GAN) artifact, codebook-based BSR approaches [7, 8], inspired by VQVAE [47, 48] and VQGAN [63], model high-resolution textures using a discrete feature space created by a pre-trained feature codebook to reconstruct HR images. These methods have shown promising results, as the codebook effectively constrains the output to fall within a valid solution space.

One of the major challenges in BSR is the complex blind degradation, which leads to similar LR versions from different HR inputs, disrupting the LR-HR matching correlation [7, 8, 21, 34, 55, 57]. For example, in Fig. 1, we sample two HR images from the DIV2K [1] dataset and degrade them using the widely used blind degradation procedure of [66]. We compute the Mean Square Error (MSE) for the similarity evaluation and find that complex degradation reduces the distinction between LR patches compared to their HR distinction. In detail, HR 1 has a smaller MSE (40.1) with LR 2, in contrast to its own corresponding LR patch (LR 1), which has an MSE of 40.3. In addition, similar LR patches tend to match the same HR patch rather than their individual HR versions. Such phenomena complicate the handling of LR data.

To address this issue, recent codebook-based methods [7, 8] incorporate an additional LR encoder to model the LR-HR relationship, based on the texture codebook learned from HR data (Fig. 2.a).

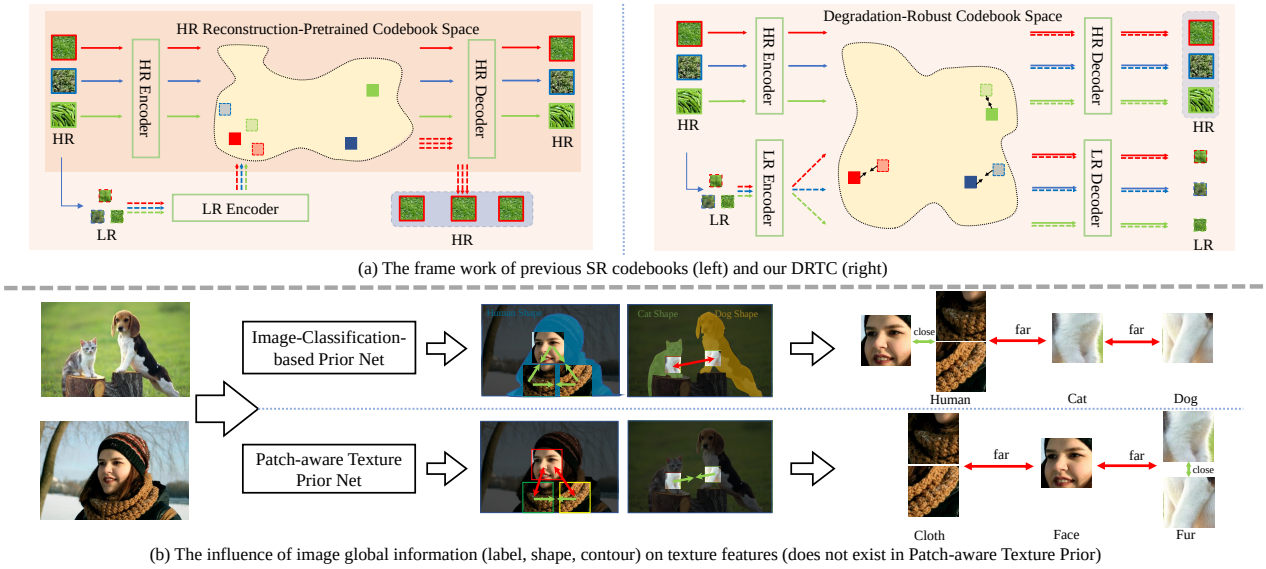


Figure 2: An illustration of our motivation. *(a)* Left: The previous HR-reconstruction-based codebook trained only on HR reconstruction, requiring a second stage training for the LR encoder; right: DTPM, which incorporates both resolutions and cross-resolution consistency during training. *(b)* Top: Image classification-based features susceptible to global factors such as class labels, object shapes, and contours; bottom: our PTPM prior without the influence of global information.

While this technique is effective when dealing with mildly degraded data, it shows lower-quality results when handling severely degraded areas. We find two main factors that limit its performance on complex degraded data. ***1)*** First, the previous codebook space, built from distinct HR data, struggles with confusing LR inputs. Unlike the clear relationship between HR textures, HR-LR relationships within BSR are confusing and often many-to-one. This poses a challenge for the previous codebooks pre-trained for HR reconstruction [7, 8] to distinguish different textures from similar degraded versions, thus limiting the diversity of texture restoration. Besides, to simplify learning, they apply the codebook only at the network bottleneck, which effectively captures larger textures but may miss mid-to-low-level details. ***2)*** Second, they use image classification-based features (often from backbones like VGG [50] pre-trained on ImageNet) for additional semantic regularization during codebook learning. However, high-level tasks that prioritize global semantics may neglect local information [22, 29] crucial for low-level tasks, causing inconsistency between pre-trained features and local texture perception (*e.g.*, Fig. 2.b). To this end, developing a texture-friendly and efficient prior based on existing global prior is worthwhile for BSR tasks, but remains underexplored.

To address the first limitation, we propose the Degradation-robust Texture Prior Module (DTPM). Unlike previous methods that rely solely on HR data for codebook learning (Fig. 2.a), DTPM involves LR data in codebook learning, improving the adaptability of codebooks to LR data. Furthermore, we exploit the distinguishability of HR representations to improve LR distinguishability by delving deeper into HR-LR correlation. Specifically, we enforce the consistency of paired HR-LR representations in codebook space and the consistency of texture content across resolutions in reconstruction results. Besides, we conduct a hierarchical codebook and a deep-to-shallow sequence training strategy for fine-grained texture modeling and stable optimization. To address the second problem, we propose the Patch-aware Texture Prior Module (PTPM) to improve the local texture perception of priors based on existing image labels. Specifically, we propose a patch-level classification-based pre-training task to reduce the global contour and shape influences. Simultaneously, we reorganize texture-friendly labels based on coarse feature clustering to correct the misleading feature similarity caused by global labels. Feature visualization and ablation studies show that PTPM offers better texture similarity assessment and benefits subsequent BSR tasks. By integrating DTPM with PTPM, we propose the Rich Texture-aware Codebook-based Network (RTCNet) for BSR. Experiments on several benchmark datasets show that RTCNet outperforms state-of-the-art methods by up to 0.16 ~ 0.46dB in PSNR and provides competitive perceptual performance. Our contributions are summarized as follows:

(1) To alleviate the limitations of previous codebook-based methods in modeling texture diversity and granularity, we propose the Degradation-robust Texture Prior Module (DTPM), which incorporates the cross-resolution consistency and hierarchical codebook structure of the texture.
(2) We propose the Patch-aware Texture Prior Module (PTPM). Compared to previous image classification-based priors, PTPM eliminates the influence of global information on local texture learning by patch-level pre-training with texture-friendly reorganized labels.
(3) Compared to recent methods, the proposed RTCNet framework combining DTPM and PTPM achieves state-of-the-art performance on multiple benchmark datasets.

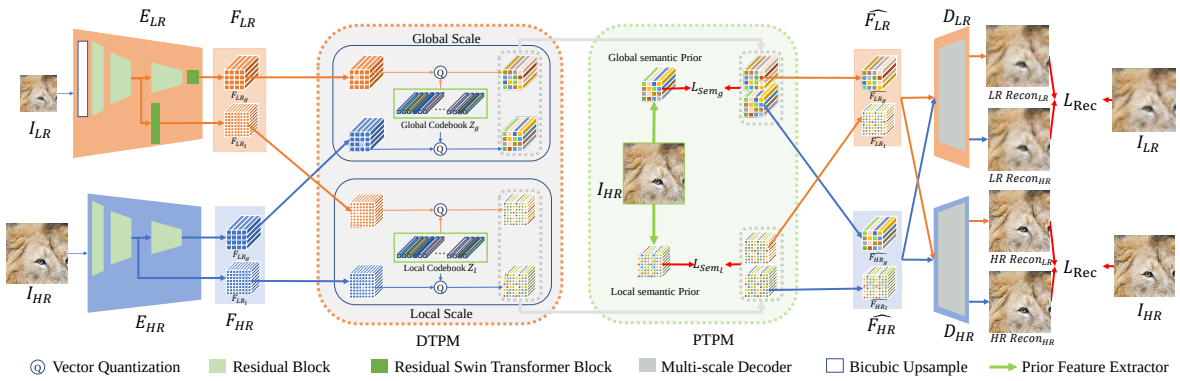


Figure 3: The RTCNet framework. (1) During training, LR and HR input images are encoded using multi-scale encoders. These features are quantized in multi-scale codebooks via DTPM. The LR and HR decoders then perform dual-resolution reconstruction. (2) During inference, only LR images are used as input; these are processed by the LR encoder and DTPM to obtain multi-scale quantized features, which are then used by the HR decoders to reconstruct super-resolution images.

2 RELATED WORK

2.1 Codebook-based SISR

Traditional codebook-based methods [5, 61] have been effective in modeling low-resolution (LR) and high-resolution (HR) patches in color spaces, especially under light degradation. However, in the case of blind super-resolution (BSR) with severe and unknown degradation, their effectiveness decreases due to the complex correspondence of different resolutions. Recent advances in deep learning [42, 47, 48, 63] have enabled the development of vector quantization-based methods [7, 8, 75] that transition patch matching from pixel to feature space, showing notable improvements in BSR scenarios. Specifically, these methods used a high-resolution VQVAE [47, 48] generation model (vector codebook and decoder) to model HR textures and an additional LR encoder for cross-resolution feature matching. Despite these advantages, as mentioned in Sec. 1, recent codebook-based methods still struggle with limited diversity and limitations and coarse modeling for fine textures. Therefore, we design the DTPM to alleviate codebook collapse and achieve hierarchical texture modeling.

2.2 Prior-based SISR

Since SR is inherently an ill-posed problem, using additional image priors can effectively improve the restoration performance. The prior-based super-resolution methods can be simply divided into explicit and implicit methods. Explicit methods [22, 30–32, 60, 71, 73, 76], which use HR reference images, can restore realistic textures but have low performance with limited reference images. Implicit methods [14, 24, 40, 44] use generative model-based priors [3] and achieve superior results on domain-specific images such as faces [4, 54, 62]. Several methods learn a posterior distribution with pre-trained StyleGAN [3] and use another encoder to project LR images into StyleGAN's latent space. However, since learning the prior from the generative model on generic images is challenging, recent methods use the high-level task-based priors for image texture reconstruction [7, 8, 59]. However, they tend to overlook local textures and instead focus on global semantics, making them less suitable for texture-sensitive image restoration.

3 METHOD

3.1 Overview

The framework of our method is shown in Fig. 3 and briefly described herein.

Training. During training, RTCNet inputs both low-resolution (LR) images I_{LR} and high-resolution (HR) images I_{HR}, with CNN encoders E_{HR}, E_{LR} used to extract hierarchical features $F_{HR}, F_{LR} = E_{HR}(I_{HR}), E_{LR}(I_{LR})$ respectively. Following prior work [8], additional RSTB [34] layers are added to E_{LR} for stronger learning ability. The extracted features are quantized via hierarchical codebooks in the Degradation-robust Texture Prior Module (DTPM),

$$\widehat{F_{HR}} = DTPM(Z, F_{HR}), \widehat{F_{LR}} = DTPM(Z, F_{LR}) \tag{1}$$

where Z denotes the hierarchical codebooks in DTPM. Finally, we pair the quantized features of all resolutions with decoders D_{HR}, D_{LR} for cross-resolution reconstruction, and compute the reconstruction loss against ground truth images.

Inference. Different from training, we only apply the LR input to the HR reconstruction process to obtain the super-resolution result I_{SR}:

$$I_{SR} = D_{HR}(DTPM(Z, E_{LR}(I_{LR})). \tag{2}$$

3.2 Degradation-robust Texture Prior Module

In this section, we introduce our Degradation-robust Texture Prior Module in detail, including vector quantization, cross-resolution consistency constraints, and the hierarchical codebook structure.

Vector Quantization. For each point feature $f \in R^C$, its quantized result $\widehat{f} \in R^C$ is the nearest neighbor based on $L2$ distance in the codebook $Z \in R^{N \times C}$ as

$$\widehat{f} = Q(Z, f) = z_m, m = \arg \min_{j \in [1,N]} ||f - z_j||_2, \tag{3}$$

where N denotes the size of the codebook. Given the input feature $F \in R^{H \times W \times C}$, its quantized feature $\widehat{F}$ is the combination of the quantized results of all point features within F, expressed as $\widehat{F} = Q(Z, F) = \{Q(Z, f_{i,j}) | i \in [1, H], j \in [1, W]\}$. Following

preious work [12, 52], we directly copy the gradient from $\widehat{F}$ to F for backpropagation and use the following loss function, L_{Code} to optimize the codebooks:

$$L_{Code}(F, \widehat{F}) = ||\widehat{F} - sg(F)||_2 + \beta \cdot ||sg(\widehat{F}) - F||_2, \tag{4}$$

where $sg(\cdot)$ means stop-gradient operation and $\beta = 0.25$ [12, 52].

In training, DTPM quantizes HR and LR features simultaneously. Its loss, L_{DTPM}, is the sum of L_{code} of hierarchical codebooks:

$$L_{DTPM} = L_{Code}(F_{HR}, \widehat{F_{HR}}) + L_{Code}(F_{LR}, \widehat{F_{LR}}). \tag{5}$$

Cross-Resolution Correlation Constraints. We investigate the texture correlation between HR and LR images, focusing on the cross-resolution consistency. We decompose the texture consistency between LR and HR data into two separate components: ***1)*** **Reconstruction consistency constraint in RGB space.** The similar code representations should have similar texture content across resolutions. Since paired HR and LR images share the same content, their quantized features $\widehat{F}_{HR}$ and $\widehat{F}_{LR}$ should be able to reconstruct both I_{LR} and I_{HR} inputs using decoders of both resolutions,

$$\begin{aligned} LR_{recon_{LR}} &= D_{LR}(\widehat{F_{LR}}), LR_{recon_{HR}} = D_{LR}(\widehat{F_{HR}}), \\ HR_{recon_{LR}} &= D_{HR}(\widehat{F_{LR}}), HR_{recon_{HR}} = D_{HR}(\widehat{F_{HR}}). \end{aligned} \tag{6}$$

Generated images should align with their corresponding resolution inputs, to which image reconstruction supervision is applied,

$$L_{Rec\ Con} = \sum_{i=\{LR,HR\}} L_{Rec}(I_{LR}, LR_{Recon_i}) + L_{Rec}(I_{HR}, HR_{Recon_i}), \tag{7}$$

where L_{Rec} denotes the image reconstruction loss function in Sec 3.4. ***2)*** **Representation consistency constraint in codebook space.** Images with similar texture content across resolutions should have similar representations in the codebook space. Specifically, we constrain the features extracted from paired HR-LR images to be consistent with each other,

$$L_{Rep\ Con} = ||F_{HR} - F_{LR}||_2. \tag{8}$$

Multi-scale Codebook Structure. The hierarchical codebook structure is based on the assumption that textures of different sizes can be characterized by codebooks of different scales. In the implementation, we employ two scales of ×4 and ×8 downsampling, hereafter referred to as local scale l and global scale g below. We apply codebooks to these scales for feature quantization,

$$\begin{aligned} DTPM(Z, F_{HR}) &= \{Q(Z_g, F_{HR_g}), Q(Z_l, F_{HR_l})\}, \\ DTPM(Z, F_{LR}) &= \{Q(Z_g, F_{LR_g}), Q(Z_l, F_{LR_l})\}. \end{aligned} \tag{9}$$

In contrast to previous bottleneck-like methods, additional shallow codebooks can represent diverse and minute texture information at smaller scales, which is helpful for generating finer textures. To mitigate convergence difficulties when training multiscale codebooks from scratch, we propose a deep-to-shallow training strategy. Specifically, codebooks are trained sequentially, starting from the deepest scales and progressing toward the shallowest scales. Fig. 4 shows the detailed training strategy. First, the global codebook is trained starting from scratch, and the temporary decoder is implemented in place of the local codebook and the multi-scale decoders. In this phase, the multi-scale encoder and the global codebook are trained well. Second, we introduce the local codebook and replace the temporary decoder with multi-scale decoders, and freeze the well-trained modules in Stage 1. The well-trained encoder and the global codebook allow for more effective and stable optimization of the local codebook.

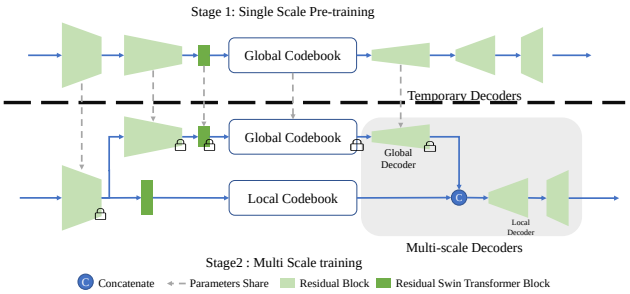


Figure 4: Hierarchical structure and its training strategy, using LR parts as an example due to the symmetry between LR and HR pipelines except for the RTSB in LR Encoder.

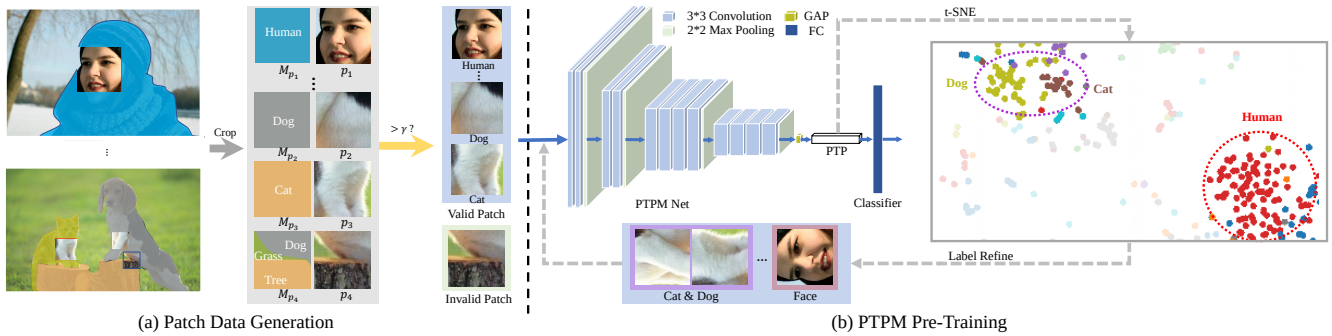
3.3 Patch-aware Texture Prior Module

To obtain a low-level friendly prior emphasizing local details over global semantics for enhanced texture perception (Sec. 1), we build our Patch-aware Texture Prior Module (PTPM) upon patch-level classification pre-training, drawing insights from multiple instance learning [45, 46] and fine-grained image classification [19]. This section details the creation process of PTPM, covering data generation and agent task pre-training.

Patch Data Generation. In general, non-overlapping patches are extracted from the images, and those without sufficient segmentation labels are discarded. The rest are assigned respective labels. Fig. 5(a) illustrates the process of extracting a patch $p \in R^{H_p \times W_p \times 3}$ from an image I with segmentation map M in a non-overlapping manner. For each patch p, we consider it valid and assign its category label Y_p as $y \in Y$, if the proportion of M_p belonging to y exceeds γ. Otherwise, it is deemed invalid,

$$VALID(p, M_p, Y) = \begin{cases} 1, & \exists\, y \in Y, \dfrac{\sum(M_p == y)}{H_p \times W_p} > \gamma, \\ 0, & \forall\, y \in Y, \dfrac{\sum(M_p == y)}{H_p \times W_p} \leq \gamma. \end{cases} \tag{10}$$

The patch-category data pairs $P = \{(p, c) | VALID(p, c) = 1\}$ are collected as the dataset for training the PTPM net. In the implementation, we manually set the threshold $\gamma = 0.85$ to balance data quality and quantity. This results in 16,818 effective patches across 27 classes, with 15,056 training and 1,763 validation samples. Dividing the images into patches allows for the separation of different texture classes. For instance, a cat or dog's head represents a characteristic patch within its class, while the body parts show inter-class similarities. As the data is cropped into patches, classifying inter-class similar patches becomes more challenging, allowing easier grouping of similar patches with different class labels while increasing the distance between class-unique and inter-class similar patches at the same time. Therefore, patch-level pre-training allows for improved texture aggregation at the patch level.



(a) Patch Data Generation (b) PTPM Pre-Training

Figure 5: PTPM consists of two main blocks: (a) patch data generation; (b) patch classification training and label refinement.

Patch Classification Pre-training. We perform patch-level classification on the collected data, as shown in Fig. 5.b. Specifically, we use the CNN part of VGG19 before the 3rd max pooling layer as our PTPM net ϕ_{patch}, pre-initialize with ImageNet pre-trained weights, and add an additional linear classifier C for pre-training. To ensure compatibility of the learned prior with the $L2$ distance in the codebook space, we add contrastive supervision $L_{InfoNCE}$ [43] to the cross-entropy loss L_{CE}. Patches within the same category are treated as positive samples, while patches from different categories are negative samples in $L_{InfoNCE}$. Given patch samples $P = \{p_i | i = 0, 1, ..., k\}$ and labels $Y = \{y_i | i = 1, 2, 3, ..., k\}$, the total prior training loss function is:

$$\begin{aligned} L_{prior} &= L_{CE} + \lambda L_{InfoNCE} \\ &= -\sum_{i=1}^{k} y_i \log(\hat{y}_i) - \lambda \log \frac{\exp(||q_i - q_{i+}||_2/\tau)}{\sum_{i=1}^{k} \exp(||q_i - q_i||_2/\tau)}, \end{aligned} \tag{11}$$

where $q_i = \phi_{patch}(p_i)$ denotes the feature embedding of p_i after GAP and $\hat{y}_i = C(q_i)$ denotes the prediction results.

Texture-orient Label Reorganization and Prior Refinement. Coarse pre-training using patch data and original image-level labels may be affected by global label influence. We mitigate this problem by reorganizing class labels based on coarse pre-training results. This process, shown in the right part of Fig. 5, entailed feature visualization by t-SNE [53], merging similar texture data with different labels, and separating discrete clusters. To further expand our data, we combined an edge-sensitive image matting dataset, resulting in 20,181 samples assigned with 35 reorganized labels. We then fine-tuned the PTPM Net using this restructured data for the final PTPM. For an intuitive comparison, we show the feature distribution comparison of our PTPM and the image classification-based prior in the appendix (Fig. 13). The PTPM feature shows better clustering performance than the image classification-based feature, signifying its increased sensitivity to texture changes. Additionally, in Fig. 6, the L2 nearest neighbors of selected samples in different prior spaces show that PTPM's method of measuring texture similarity more closely aligns with human perception.

3.4 Training losses

Codebook Loss. This loss optimizes DTPM, including the codebook loss and two correlation constraints:

$$L_{Codebook} = L_{DTPM} + L_{Rep\ Con} + L_{Rec\ Con}. \tag{12}$$

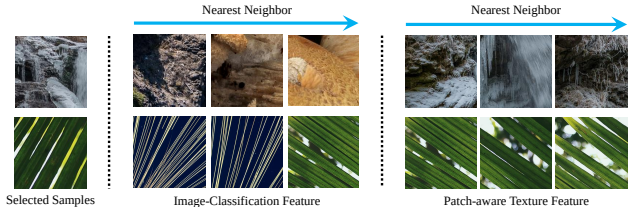


Figure 6: L2 nearest neighbors of several selected samples in different prior spaces.

Image Reconstruction Loss. We use $L1$ and Perceptual Loss [23] as the main reconstruction loss. Following previous work [7, 8], we use a U-net discriminator D in [55] and a hinge loss as an adversarial loss to get more realistic textures. Given a reconstructed image I_{Recon} and its ground truth image I_{GT}, the image reconstruction loss can be formulated as

$$\begin{aligned} L_{Rec}(I_{GT}, I_{Recon}) =& ||I_{GT} - I_{Recon}||_1 \\ &+ \lambda_{per} ||\phi_{per}(I_{GT}) - \phi_{per}(I_{Recon})||_1 + \lambda_{adv} E[D(I_{Recon})], \end{aligned} \tag{13}$$

where ϕ_{per} denotes a pre-trained VGG16 [50] network.

PTPM Loss. We integrate the PTPM prior into the DTPM's training by applying scale-matched texture prior regularization. Specifically, the global texture priors are the activations of the 5th ReLU of the ImageNet-pretrained VGG19 [50] network ϕ_{img} and the local-friendly priors are the activations of the 2nd Max Pooling of our PTPM Net ϕ_p. We extract the texture priors from the HR images. The PTPM supervision L_{PTPM} is computed between the quantized features $\widehat{F}$ and the corresponding texture priors which can be formulated as

$$L_{PTPM}(I_{HR}, \widehat{F}) = ||\phi_{img}(I_{HR}) - \phi_{p_g}(\widehat{F_g})||_2 + ||\phi_p(I_{HR}) - \phi_{p_l}(\widehat{F_l})||_2, \tag{14}$$

where $\phi_{p_{l,g}}$ are single convolution layer to transfer from the codebook space to the prior space. The total PTPM supervision is the sum of the supervision on the quantized features of two scales as

$$L_{PTPM} = L_{PTPM}(I_{HR}, \widehat{F_{HR}}) + L_{PTPM}(I_{HR}, \widehat{F_{LR}}). \tag{15}$$

Overall Loss. The overall loss is then defined as

$$L_{total} = L_{Codebook} + L_{PTPM}. \tag{16}$$

Table 1: Quantitative comparison (PSNR ↑, SSIM ↑) with state-of-the-art BSR methods on 6 different benchmarks.

Method	DIV2K		Urban100		BSDS100		Manga109		Set14		Set5	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CDC (2020)	19.79	0.4735	17.43	0.4010	20.13	0.4384	17.64	0.5223	19.69	0.4802	18.90	0.4717
DAN (2020)	20.07	0.4577	17.74	0.4034	20.46	0.4341	18.13	0.5229	19.83	0.4727	19.63	0.4697
Real-ESRGAN (2021b)	20.08	0.5273	17.51	0.4443	20.31	0.4383	18.76	0.6064	20.05	0.4723	19.95	0.5125
SwinIR-GAN (2021)	20.37	0.5283	17.64	0.4562	20.15	0.4310	19.18	**0.6251**	20.21	0.4776	19.44	0.4680
BSRDM (2022)	20.19	0.5330	17.18	0.4031	19.94	0.4333	17.31	0.5437	19.17	0.4621	18.62	0.4715
D2C-SR (2022b)	19.44	0.4156	17.40	0.3801	20.12	0.4199	17.54	0.4933	19.82	0.4765	18.61	0.4229
KXNet (2022)	20.10	0.4696	17.57	0.3992	20.34	0.4341	17.84	0.5214	19.64	0.4702	19.28	0.4650
MM-RealSR (2022)	20.60	**0.5471**	17.95	0.4585	20.34	0.4473	18.80	0.6153	20.02	0.4817	19.84	**0.5152**
FeMaSR (2022b)	20.31	0.4918	18.01	0.4384	20.09	0.4156	19.15	0.6024	20.18	0.4581	19.57	0.4536
MRDA (2023)	19.91	0.4474	17.70	0.4009	20.43	0.4328	18.07	0.5202	19.82	0.4731	19.57	0.4666
RTCNet(ours)	**20.76**	0.5268	**18.40**	**0.4586**	**20.91**	**0.4537**	**19.52**	0.6133	**20.38**	**0.4829**	**20.32**	0.4931

Table 2: Perceptual metrics (LPIPS [67] ↓) comparison with state-of-the-art blind super-resolution methods on DIV2K.

Methods	CDC	DAN	SwinIR	Real-ESRGAN	BSRDM	MM-RealSR	KXNet	D2C-SR	FeMaSR	MRDA	RTCNet(ours)
LPIPS	0.7722	0.7466	0.4739	0.5637	0.7505	0.5724	0.7655	0.7689	0.4480	0.7464	**0.4390**

4 EXPERIMENTS

4.1 Datasets and Evaluation Metrics

Prior Pre-training Datasets. Our coarse patch classification dataset is based on the ADE20K [74] semantic segmentation dataset and expanded using the SIM [51] image matting dataset. Following the strategy in Sec 3.3, we generate a final dataset with $17,880$ training samples and $2,301$ validation samples.

Super-Resolution Training Dataset. We build an overall training dataset including DV2K [1], DIV8K [15], Flickr2K [35], and OST [59] datasets. HR patches are generated using the following approach: *1)* crop large images into non-overlapping 512×512 patches; *2)* apply the blur detection method [25] to each patch to filter out blurred patches with a blurred area greater than 95%. Our final training dataset contains $123,395$ HR patches, while we generated LR patches for each iteration using the widely used degradation model proposed in [66].

Super-Resolution Test Datasets. We evaluated the performance of our model using six benchmark datasets, namely DIV2K [1], Set14 [65], Set5 [2], Urban100 [20], BSD100 [38], and Manga109 [39]. We used the mixed degradation model described in [55] and [66] for LR generation. The diverse datasets with complex degradation, allowed for a comprehensive performance evaluation. A 4x downsampling was used for all experiments.

Evaluation Metrics. We used Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) to evaluate the quality of generated images. In addition, for better perceptual evaluation, we also use the Learned Perceptual Image Patch Similarity (LPIPS) [67].

Implementation Details. We implement our model using the PyTorch framework. In both low-level prior pretraining and SR training, we use an Adam [27] optimizer with $\beta_1 = 0.9, \beta_2 = 0.99, lr = 1 \times 10^{-4}$. The number of codes in both scale codebooks is set to 512. The RTCNet is trained with a batch size of 16 and a HR patch size of 256×256 on 4 NVIDIA V100 GPUs for about 4 days.

4.2 Comparison with SOTA

We compared the proposed RTCNet with 10 recent state-of-the-art blind SR methods, including CDC [57], DAN [21], Real-ESRGAN [55], SwinIR-GAN [34], BSRDM [64], D2C-SR [33], KXNet [13], MM-RealSR [41], FeMaSR [8] and MRDA [58]. We compare our method with these approaches using the published codes and weights from the official public GitHub repos.

As shown in Tab. 1, our method achieves the best PSNR/SSIM performance on almost all 6 datasets. In Fig. 7, we compare the restored images of different BSR methods. First, consistent with the results in Tab. 1 and Tab. 2, DAN and CDC have limited recovery effects for complex degraded images. Second, Real-ESRGAN and SwinIR-GAN tend to confuse noise and texture. They erase texture details as noise and cause over-smoothing problems. Besides, FeMaSR mistakes some noise for texture, resulting in noisy texture generation. On the contrary, since our DTPM effectively maintains the cross-resolution consistency of texture codebooks, it is more robust to low-resolution degradation. It can reasonably distinguish between texture and degradation, ensuring the restoration of realistic textures while denoising. The multi-scale structure and low-level friendly priors further improve the restoration of local fine textures. In general, our RTCNet achieves state-of-the-art performance in quantitative metrics and human perception.

4.3 Ablation Study

Effectiveness of Cross-Resolution Correlation. To verify the effectiveness of cross-resolution constraints, we conduct an ablation study on the two cross-resolution strategies used: cross-resolution representation consistency (Rep. C.) and cross-resolution reconstruction consistency (Rec. C.). As shown in Tab. 3, both of them can effectively improve the performance of DTPM. This is because

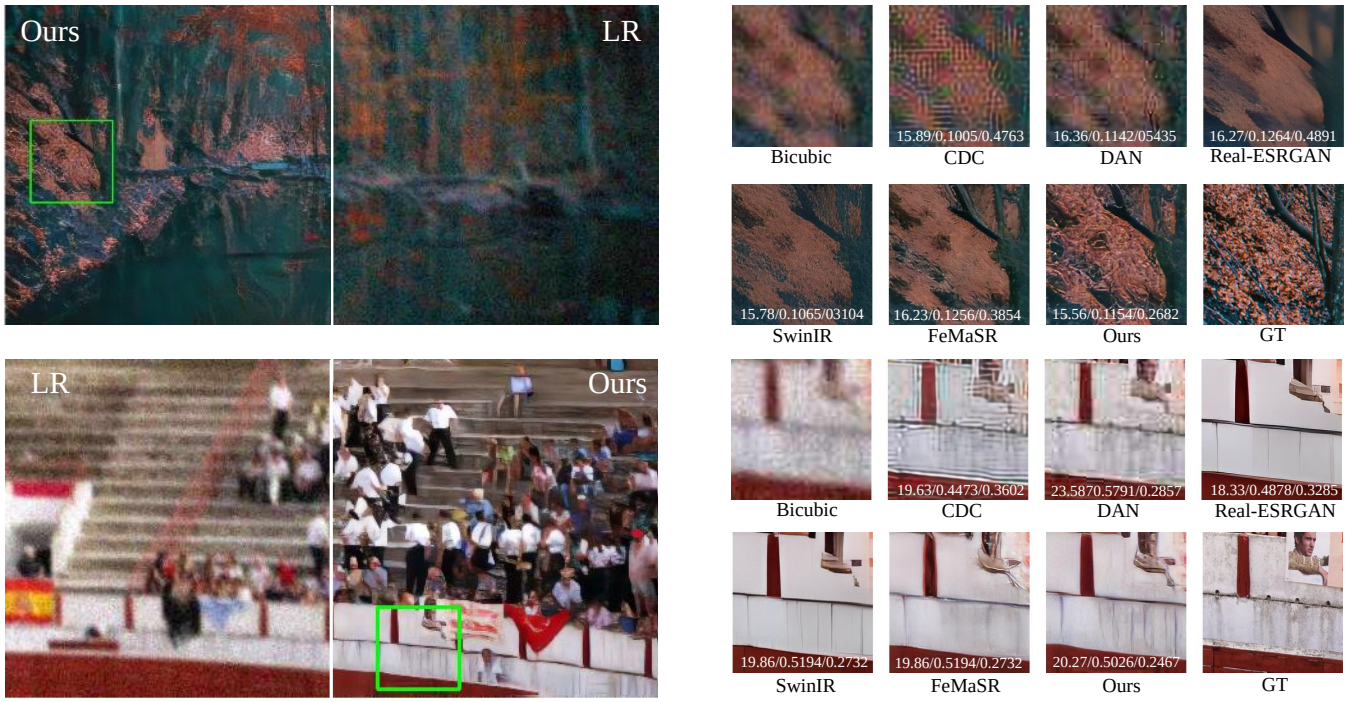


Figure 7: Visual comparison with other blind super-resolution images. The PSNR/SSIM/LPIPS values are shown at the bottom of the images. The captions in the images below have the same meaning as the descriptions provided here.

Table 3: Ablation of DTPM. Rows 1-4: Ablation of DTPM consistency constraints. Rows 5-7: Ablation of components in hierarchical codebook learning. Rep. C.: Representation Consistency; Rec. C.: Reconstruction Consistency. H.S.: Hierarchical Structure; D2S: Deep-to-Shallow strategies.

Rep. C.	Rec. C.	H.S.	D2S	PTPM	PSNR	SSIM
×	×	×	-	-	19.90	0.5004
×	✓	×	-	-	20.17	0.4974
✓	×	×	-	-	20.19	0.4990
✓	✓	×	-	-	20.59	0.5180
✓	✓	✓	×	×	19.53	0.4781
✓	✓	✓	✓	×	20.52	0.5215
✓	✓	✓	✓	✓	**20.76**	**0.5268**

the cross-resolution constraint forces the LR representation in the codebook to be closer to the HR, making it as distinguishable as the HR in the codebook space. And the combination of the two can further enhance the improvement.

Effectiveness of Hierarchical Structure. We validated the effectiveness of prior feature regularization and deep-to-shallow training strategy for multi-scale codebook training in Tab. 3. Training a multi-scale model from scratch leads to insufficient texture learning due to the more diverse and sensitive texture degradation at the local scale, making its performance even worse than that of the single-scale model (see rows 4 and 5, Tab. 3). The addition of the deep-to-shallow training strategy stabilizes the learning of the local scale codebook by the well-trained encoders and global features, which significantly improves the restoration of large textures. Notably, the performance of the deep-to-shallow strategy is not significantly better than the single-scale model, while after adding the PTPM regularization on this basis, the result is better than the single-scale model. This shows that the full multi-scale model can achieve better performance for fine textures than the single-scale model through hierarchical texture learning, but the learning of local-scale texture is challenging and requires the assistance of a low-level texture-friendly prior.

Table 4: Comparison of DTPM with the high-resolution reconstruction-based codebook of FeMaSR [8] in the DIV2K validation set. The '*' indicates that we conduct a single-scale codebook without the hierarchical structure.

	PSNR	SSIM	Codebook Use ratio
FeMaSR [8]	20.31	0.4918	33 / 1024
DTPM*	**20.59**	**0.5180**	396 / 512

Comparison with Previous Codebooks. To verify the superiority of our proposed DTPM, we compare it with the high-resolution reconstruction-based codebook of FeMaSR [8]. For fairness, both of them used the bottleneck model structure (codebook at x8 downsampling) and trained with our overall loss except for the local-scale PTPM loss. As shown in Tab. 4 and Fig. 8, DTPM outperforms FeMaSR [8] in both quantitative and qualitative results. Compared to FeMaSR [8], our single-scale DTPM has a more stable and realistic

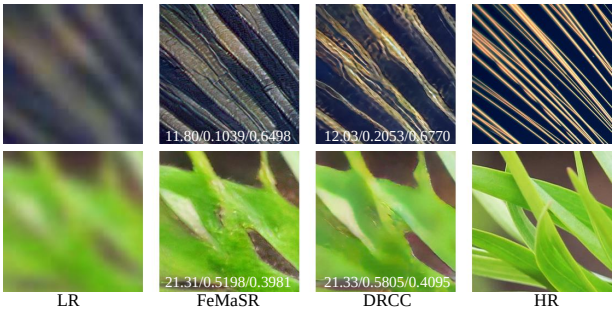


Figure 8: Visual comparison between single-scale DTPM and FeMaSR [8].

Table 5: Comparison of different priors used to learn the local-scale codebook in the DIV2K validation set. († denotes the coarse prior before label refinement and fine-tuning).

Local Prior	PSNR	SSIM
-	20.52	0.5215
ImgNet-Classification Prior	20.57	0.5240
Patch-aware Texture Prior†	20.67	**0.5314**
Patch-aware Texture Prior	**20.76**	0.5268

Figure 9: Visual comparison of local-scale codebooks trained with different prior features.

texture generation in heavy degradation. To show the advantage of DTPM more intuitively, we also statistically analyze the codebook utilization in the inference stage. As shown in Tab. 4, only 3.2% of FeMaSR's codebook was used during inference. Such an over collapse indicates the inadequacy and indistinguishability of the HR-reconstruction-based codebook when applied to LR data. This limitation of the codebook limits the variety of textures that can be generated, resulting in unrealistic recovery during super-resolution. In contrast, DTPM has a wider range of codebook usage. This is because the codebook space is trained with LR data and contains more discriminative LR representations under the cross-resolution consistency constraint. Benefiting from this, DTPM achieves more diverse texture generation and stronger stability (less noise on row 2 and clear texture on row 1 in Fig. 8).

Effectiveness of PTPM in blind super-resolution. To verify the superiority of our PTPM for low-level texture learning, we compared the impact of different semantic features on the learning of the local codebook in RTCNet. As shown at the bottom of Fig. 9 and in Tab. 5, while all pre-trained priors improve the texture restoration performance, our PTPM prior outperforms the ImageNet-based pre-trained prior. This superiority can be attributed to our PTPM's better perception of low-level texture correlations. In addition, patch-level pre-training and texture-oriented label organization both improve the performance of the PTPM. To demonstrate the superior ability of the PTPM prior to distinguish different types of textures compared to the image classification-based prior, we analyzed the frequency distribution of different codes used for super-resolution on the OST dataset in the appendix (Fig. 14). As can be observed, compared to the ImageNet Classification pre-trained priors, PTPM shows more distribution differences between the "grass" and "plant" categories, which have more overlapping semantic labels, and has a smaller difference in the "sky" and "water" categories, which have different semantic categories but relatively similar textures. This shows that our pre-training strategy enables PTPMNet to pay more attention to the correlation of local texture information by excluding high-level information from the pre-training.

4.4 Limitation and Discussion

First, by observing the results, we find that RTCNet has some limitations when dealing with regular texture restoration, especially for data types that have plenty of such textures, such as buildings (examples in Supplement). This problem also occurs with the previous codebook-based method, which we will investigate in future work. Second, we find that the improvement of RTCNet is more obvious in the heavily degraded samples than in the lightly degraded samples (perhaps no improvement in some light samples) (Fig. 11.c). We speculate that the notable improvements in the heavily degraded data are due to increased matching confusion, a scenario where RTCNet performs optimally. Conversely, light degradation with less confusion can also be handled by previous methods, leading to marginal improvements. Although both data types are common in applications, we argue that the correction of complex degraded data has great challenges and value for super-resolution (SR) tasks. Third, the improvements brought by PTPM are not very considerable and stable, indicating that larger valid datasets and a more refined pre-training strategy is valuable for better performance. Furthermore, based on experience, pre-training features tend to adapt more effectively to data with strong domain priors, suggesting that applying the codebook method in combination with pre-training strategies to specific types of data may be a direction worth investigating.

5 CONCLUSION

In this paper, we have presented the Rich Texture-aware Codebook Network (RTCNet) framework for blind image super-resolution. With our proposed Degradation-aware Texture Codebook Module, we allow for more efficient modeling of LR-HR correspondences than previous single HR reconstruction pre-training. The architectures of DTPM allow it to model large and fine textures separately. In addition, we build the low-level friendly Patch-aware Texture Prior Module (PTPM) which further improves the performance of DTPM. Various experiments on different benchmarks show that our RTCNet achieves state-of-the-art performance.

REFERENCES

[1] Eirikur Agustsson and Radu Timofte. 2017. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 1122–1131. https://doi.org/10.1109/CVPRW.2017.150

[2] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. 2012. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. (2012).

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

[4] Kelvin CK Chan, Xintao Wang, Xiangyu Xu, Jinwei Gu, and Chen Change Loy. 2021. Glean: Generative latent bank for large-factor image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 14245–14254.

[5] Hong Chang, Dit-Yan Yeung, and Yimin Xiong. 2004. Super-resolution through neighbor embedding. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, Vol. 1. IEEE, I–I.

[6] Chaofeng Chen, Dihong Gong, Hao Wang, Zhifeng Li, and Kwan-Yee K Wong. 2020. Learning spatial attention for face super-resolution. *IEEE Transactions on Image Processing* 30 (2020), 1219–1231.

[7] Chaofeng Chen, Xinyu Shi, Yipeng Qin, Xiaoming Li, Xiaoguang Han, Tao Yang, and Shihui Guo. 2022. Blind Image Super Resolution with Semantic-Aware Quantized Texture Prior. *arXiv preprint arXiv:2202.13142* (2022).

[8] 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*. 1329–1338.

[9] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. 2021. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12299–12310.

[10] Lei Ding, Hao Tang, and Lorenzo Bruzzone. 2021. LANet: Local Attention Embedding to Improve the Semantic Segmentation of Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing* 59, 1 (2021), 426–435. https://doi.org/10.1109/TGRS.2020.2994150

[11] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2014. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*. Springer, 184–199.

[12] Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12873–12883.

[13] Jiahong Fu, Hong Wang, Qi Xie, Qian Zhao, Deyu Meng, and Zongben Xu. 2022. KXNet: A model-driven deep neural network for blind super-resolution. In *European Conference on Computer Vision*. Springer, 235–253.

[14] Jinjin Gu, Yujun Shen, and Bolei Zhou. 2020. Image processing using multi-code gan prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3012–3021.

[15] Shuhang Gu, Andreas Lugmayr, Martin Danelljan, Manuel Fritsche, Julien Lamour, and Radu Timofte. 2019. DIV8K: DIVerse 8K Resolution Image Dataset. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. 3512–3516. https://doi.org/10.1109/ICCVW.2019.00435

[16] Baisong Guo, Xiaoyun Zhang, Haoning Wu, Yu Wang, Ya Zhang, and Yan-Feng Wang. 2022. LAR-SR: A Local Autoregressive Model for Image Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1909–1918.

[17] Kaiming He, Ross Girshick, and Piotr Dollár. 2019. Rethinking imagenet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4918–4927.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[19] Le Hou, Dimitris Samaras, Tahsin M Kurc, Yi Gao, James E Davis, and Joel H Saltz. 2016. Patch-based convolutional neural network for whole slide tissue image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2424–2433.

[20] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. 2015. Single Image Super-Resolution From Transformed Self-Exemplars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[21] Yan Huang, Shang Li, Liang Wang, Tieniu Tan, et al. 2020. Unfolding the alternating optimization for blind super resolution. *Advances in Neural Information Processing Systems* 33 (2020), 5632–5643.

[22] Yuming Jiang, Kelvin CK Chan, Xintao Wang, Chen Change Loy, and Ziwei Liu. 2021. Robust reference-based super-resolution via C2-matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2103–2112.

[23] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*. Springer, 694–711.

[24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8110–8119.

[25] Beomseok Kim, Hyeongseok Son, Seong-Jin Park, Sunghyun Cho, and Seungyong Lee. 2018. Defocus and motion blur detection with deep contextual features. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 277–288.

[26] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. 2016. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1646–1654.

[27] P Kingma Diederik and Jimmy Ba Adam. 2014. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[28] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4681–4690.

[29] Tianhong Li, Huiwen Chang, Shlok Kumar Mishra, Han Zhang, Dina Katabi, and Dilip Krishnan. 2022. Mage: Masked generative encoder to unify representation learning and image synthesis. *arXiv preprint arXiv:2211.09117* (2022).

[30] Xiaoming Li, Chaofeng Chen, Shangchen Zhou, Xianhui Lin, Wangmeng Zuo, and Lei Zhang. 2020. Blind face restoration via deep multi-scale component dictionaries. In *European Conference on Computer Vision*. Springer, 399–415.

[31] Xiaoming Li, Wenyu Li, Dongwei Ren, Hongzhi Zhang, Meng Wang, and Wangmeng Zuo. 2020. Enhanced blind face restoration with multi-exemplar images and adaptive spatial feature fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2706–2715.

[32] Xiaoming Li, Ming Liu, Yuting Ye, Wangmeng Zuo, Liang Lin, and Ruigang Yang. 2018. Learning warped guidance for blind face restoration. In *Proceedings of the European conference on computer vision (ECCV)*. 272–289.

[33] Youwei Li, Haibin Huang, Lanpeng Jia, Haoqiang Fan, and Shuaicheng Liu. 2022. D2c-sr: A divergence to convergence approach for real-world image super-resolution. In *European Conference on Computer Vision*. Springer, 379–394.

[34] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1833–1844.

[35] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. 2017. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 136–144.

[36] Guandu Liu, Yukang Ding, Mading Li, Ming Sun, Xing Wen, and Bin Wang. 2023. Reconstructed Convolution Module Based Look-Up Tables for Efficient Image Super-Resolution. *arXiv preprint arXiv:2307.08544* (2023).

[37] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tieyong Zeng. 2022. Transformer for Single Image Super-Resolution. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 456–465. https://doi.org/10.1109/CVPRW56347.2022.00061

[38] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, Vol. 2. IEEE, 416–423.

[39] Yusuke Matsui, Kota Ito, Yuji Aramaki, Azuma Fujimoto, Toru Ogawa, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2017. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications* 76 (2017), 21811–21838.

[40] Sachit Menon, Alexandru Damian, Shijia Hu, Nikhil Ravi, and Cynthia Rudin. 2020. Pulse: Self-supervised photo upsampling via latent space exploration of generative models. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*. 2437–2445.

[41] Chong Mou, Yanze Wu, Xintao Wang, Chao Dong, Jian Zhang, and Ying Shan. 2022. Metric learning based interactive modulation for real-world super-resolution. In *European Conference on Computer Vision*. Springer, 723–740.

[42] Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, Kaihao Zhang, Xiaochun Cao, and Haifeng Shen. 2020. Single image super-resolution via a holistic attention network. In *European conference on computer vision*. Springer, 191–207.

[43] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[44] Xingang Pan, Xiaohang Zhan, Bo Dai, Dahua Lin, Chen Change Loy, and Ping Luo. 2021. Exploiting deep generative prior for versatile image restoration and manipulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[45] Deepak Pathak, Evan Shelhamer, Jonathan Long, and Trevor Darrell. 2014. Fully convolutional multi-class multiple instance learning. *arXiv preprint arXiv:1412.7144* (2014).

[46] Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2015. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In *Proceedings of the 2015 conference on empirical methods in natural language processing*. 2539–2544.

[47] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. 2019. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems* 32 (2019).

[48] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. 2019. Generating diverse high-resolution images with VQ-VAE. (2019).

[49] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science* 290, 5500 (2000), 2323–2326.

[50] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[51] Yanan Sun, Chi-Keung Tang, and Yu-Wing Tai. 2021. Semantic Image Matting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 11120–11129.

[52] Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems* 30 (2017).

[53] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html

[54] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. 2021. Towards real-world blind face restoration with generative facial prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9168–9178.

[55] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. 2021. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1905–1914.

[56] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. 2018. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*. 0–0.

[57] Pengxu Wei, Ziwei Xie, Hannan Lu, Zongyuan Zhan, Qixiang Ye, Wangmeng Zuo, and Liang Lin. 2020. Component divide-and-conquer for real-world image super-resolution. In *European Conference on Computer Vision*. Springer, 101–117.

[58] Bin Xia, Yapeng Tian, Yulun Zhang, Yucheng Hang, Wenming Yang, and Qingmin Liao. 2023. Meta-Learning-Based Degradation Representation for Blind Super-Resolution. *IEEE Transactions on Image Processing* 32 (2023), 3383–3396. https://doi.org/10.1109/TIP.2023.3283922

[59] Chao Dong Xintao Wang, Ke Yu and Chen Change Loy. 2018. Recovering Realistic Texture in Image Super-resolution by Deep Spatial Feature Transform. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[60] 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.

[61] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. 2010. Image super-resolution via sparse representation. *IEEE transactions on image processing* 19, 11 (2010), 2861–2873.

[62] Tao Yang, Peiran Ren, Xuansong Xie, and Lei Zhang. 2021. Gan prior embedded network for blind face restoration in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 672–681.

[63] Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. 2021. Vector-quantized image modeling with improved vqgan. *arXiv preprint arXiv:2110.04627* (2021).

[64] Zongsheng Yue, Qian Zhao, Jianwen Xie, Lei Zhang, Deyu Meng, and Kwan-Yee K Wong. 2022. Blind image super-resolution with elaborate degradation modeling on noise and kernel. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2128–2138.

[65] Roman Zeyde, Michael Elad, and Matan Protter. 2012. On single image scale-up using sparse-representations. In *Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7*. Springer, 711–730.

[66] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. 2021. Designing a practical degradation model for deep blind image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4791–4800.

[67] 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*. 586–595.

[68] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. 2018. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*. 286–301.

[69] Yulun Zhang, Kunpeng Li, Kai Li, Bineng Zhong, and Yun Fu. 2019. Residual non-local attention networks for image restoration. *arXiv preprint arXiv:1903.10082* (2019).

[70] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. 2018. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2472–2481.

[71] Zhifei Zhang, Zhaowen Wang, Zhe Lin, and Hairong Qi. 2019. Image super-resolution by neural texture transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7982–7991.

[72] Kai Zhao, Kun Yuan, Ming Sun, Mading Li, and Xing Wen. 2023. Quality-aware pre-trained models for blind image quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 22302–22313.

[73] Haitian Zheng, Mengqi Ji, Haoqian Wang, Yebin Liu, and Lu Fang. 2018. Crossnet: An end-to-end reference-based super resolution network using cross-scale warping. In *Proceedings of the European conference on computer vision (ECCV)*. 88–104.

[74] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2017. Scene Parsing through ADE20K Dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5122–5130. https://doi.org/10.1109/CVPR.2017.544

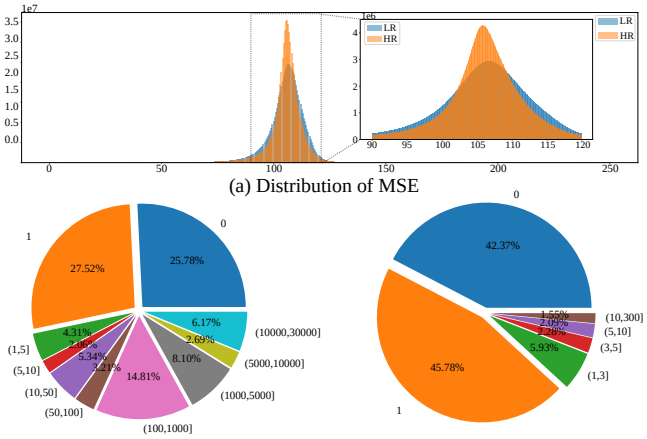
[75] Shangchen Zhou, Kelvin Chan, Chongyi Li, and Chen Change Loy. 2022. Towards robust blind face restoration with codebook lookup transformer. *Advances in Neural Information Processing Systems* 35 (2022), 30599–30611.

[76] Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, and Chen Change Loy. 2020. Cross-scale internal graph neural network for image super-resolution. *Advances in neural information processing systems* 33 (2020), 3499–3509.

A APPENDIX

A.1 LR Confusion in BSR data

This section presents a statistical analysis of the LR data from all validation datasets used to show the universal confusion phenomenon observed in LR data compared to HR data. We densely cropped all HR images and their corresponding LR versions, which have the same size as the HR version after bicubic upsampling, into 128×128 patches (a total of 26,753 patches). We then computed the mean squared error (MSE) between all HR and LR patches. First, we analyzed the distributions of both HR and LR patches using the MSE, as shown in Figure 10.a. The plot highlights that HR patches have a more concentrated MSE distribution, while LR patches have a more dispersed one. This indicates that the LR data are more prone to confusion. Second, we extracted the index of HR patches in the nearest HR patch sorting of its LR patch. As shown in Figure 10.b, the index is dispersed, with a significant proportion not being the top-1 nearest. This implies that many LR patches have a closer MSE distance to other HR patches than their corresponding HR patches. Furthermore, Fig. 10.c shows the frequency of selection of each HR patch as the nearest one to different LR patches. The figure shows that there is a large partition of non-1 frequency, indicating a large part of the LR-HR mismatch. Although the MSE statistic is not entirely suitable for evaluating the similarity between patches, the considerable partition of mismatches between HR patches and their LR counterparts suggests the confusion caused by blind degradation and the complex correlation it introduces.



(a) Distribution of MSE

(b) Distribution of HR's Nearest Index

(c) Distribution of HR's Top-1 Nearest Frequency

Figure 10: The MSE statistics of LR-HR data in validation datasets.

A.2 Detailed Comparison between DTPM and FeMaSR

In this section, we performed a comparative analysis of our DTPM and previous high-resolution reconstruction-based codebooks (using FeMaSR as an example). We statistically investigated the performance improvements of DTPM over the FeMaSR method on samples of varying difficulty in the DIV2K validation set. Specifically, we divided the high-resolution (HR), low-resolution (LR), and their super-resolution (SR) results into 128×128 patches(15585 in total). We used the mean squared error (MSE) distance between LR and HR as a simple measure of sample difficulty, with smaller values indicating easier samples and larger values indicating more difficult samples. We compared and plotted the measurements including MSE(Fig. 11.a), Peak Signal to Noise Ratio (PSNR, Fig. 11.b), and Structural Similarity Index (SSIM, Fig. 11.d) of the SRs of DTPM and FeMaSR under different levels of difficulty. To better illustrate the advantages of DTPM on difficult samples, we also investigate the performance gain of DTPM over the FeMaSR method for different sample difficulty levels(Fig. 11.c). As shown in Fig. 11, compared to FeMaSR, our DTPM has achieved improvements in different levels of difficulty, especially for samples with higher difficulty Tab. 11.c. This verifies the good adaptability of DTPM to LR data, and thanks to its mining of texture cross-resolution consistency, DTPM can better distinguish different types of textures and perform diverse reconstructions for more difficult samples.

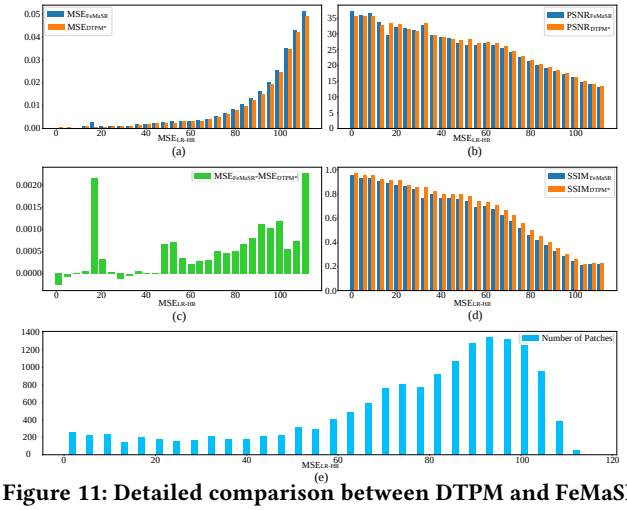


Figure 11: Detailed comparison between DTPM and FeMaSR. (a) Distribution of MSE between LR and HR. (b) Distribution of PSNR between SR and HR. (c) Distribution of PSNR gain of DTPM over FeMaSR. (d) Distribution of SSIM between SR and HR. (e) Number distribution of image patches with different LR-HR MSE.

A.3 Validation of Hierarchical texture learning of multi-scale structure

To better understand the advantage of hierarchical codebooks for texture learning, we explore the texture content learned at different scales in the hierarchical codebook architecture. Specifically, during the quantization process of local-scale DTPM, we replace the quantized features obtained from the low-resolution input with the noise features generated by the random indexes, thereby removing the influence of the local scale feature during the reconstruction process. The quantitative and qualitative results are shown in Tab. 6 and Fig. 12, respectively.

In Fig. 12, when the local-scale information is missing, detailed texture restoration is heavily affected, causing unrealistic fine texture reconstruction. In contrast, the global contour and large-scale

Table 6: Comparison between full hierarchical structure DTPM and noisy-local DTPM.

Method	PSNR	SSIM
RTCNet(noisy local-scale code)	20.17	0.4928
RTCNet	**20.76**	**0.5268**

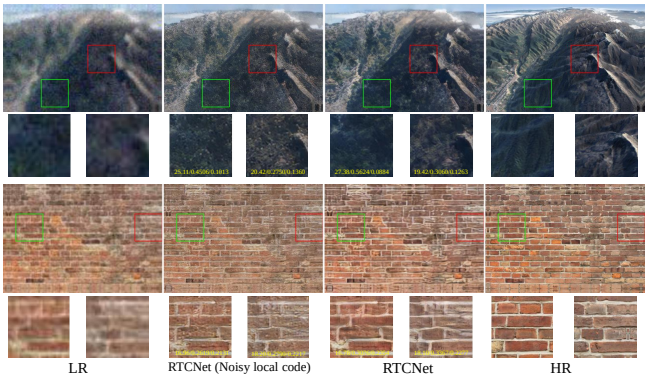


Figure 12: Reconstruction Comparison of the local-scale quantized features generated by random noise and the matching local-scale quantized features obtained from the input (more samples in the Supplement).

textures are not significantly affected. This shows that the hierarchical structure learns textures of different sizes with different-scale codebooks. The global codebook and local codebook are responsible for global and local-scale textures separately. Such a strategy improves the model's ability to model different textures and increases the diversity of textures in the reconstruction results.

A.4 More Analysis Experiment of PTPM Prior Features

We present the detailed visualized comparison using t-SNE dimensionality reduction between image classification-based priors and our PTPM priors in Fig. 13. Compared to image classification-based priors, our PTPM priors have better clustering performance, indicating a higher sensitivity to local texture similarity. To further illustrate the difference between our PTPM prior and the ImageNet prior in the process of learning low-level texture, we conducted super-resolution statistics on the OST dataset. The high-resolution images in the OST dataset were divided into seven categories according to rough textures, including animal, building, plant, grass, sky, water, and mountain. We degraded the HR data in the OST dataset and perform BSR on them. Then we counted the usage frequency of each code in the codebook during the super-resolution process by category. By comparing the distribution of codes used when facing different textures, we compare the rationality of learned code spaces for texture perception. As can be observed in Fig. 14, compared to the ImageNet Classification pre-trained priors, PTPM has more distribution differences between the "grass" and "plant" categories, which have more overlapping semantic labels, and has a smaller difference in the "sky" and "water" categories, which have different semantic categories but relatively similar textures. This shows that our pre-training strategy allows PTPMNet to pay more attention to the correlation of local texture information by excluding high-level information from the pre-training.

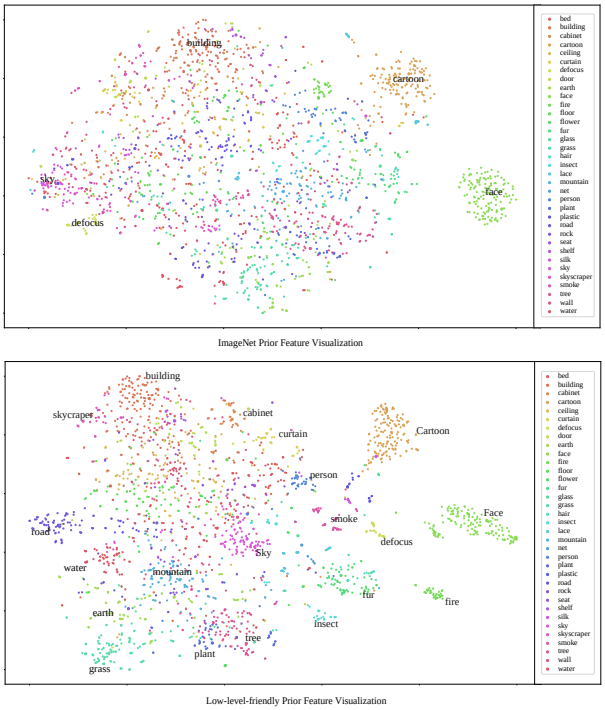


Figure 13: The detailed t-SNE visualization of different prior features extracted from the images of our low-level patch classification validation dataset with the legend of patch-classification dataset classes.

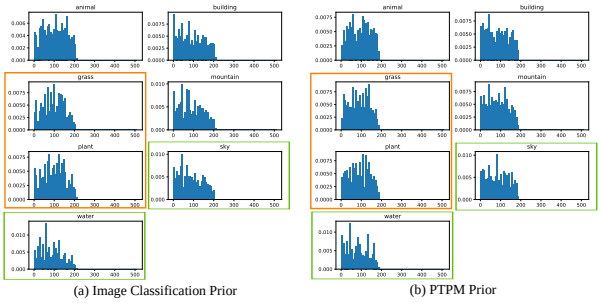


Figure 14: The frequency distribution of different codes used during super-resolution on the OST dataset.