LAR-SR: A Local Autoregressive Model for Image Super-Resolution

Baisong Guo[1*], Xiaoyun Zhang[1*†], Haoning Wu[1], Yu Wang[1,2], Ya Zhang[1,2], Yan-Feng Wang[1,2 †]
[1]Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, [2]Shanghai AI Laboratory
{stilltooyoung,xiaoyun.zhang,whn15698781666,yuwangsjtu,ya_zhang,wangyanfeng}@sjtu.edu.cn

Abstract

Previous super-resolution (SR) approaches often formulate SR as a regression problem and pixel wise restoration, which leads to a blurry and unreal SR output. Recent works combine adversarial loss with pixel-wise loss to train a GAN-based model or introduce normalizing flows into SR problems to generate more realistic images. As another powerful generative approach, autoregressive (AR) model has not been noticed in low level tasks due to its limitation. Based on the fact that given the structural information, the textural details in the natural images are locally related without long term dependency, in this paper we propose a novel autoregressive model-based SR approach, namely LAR-SR, which can efficiently generate realistic SR images using a novel local autoregressive (LAR) module. The proposed LAR module can sample all the patches of textural components in parallel, which greatly reduces the time consumption. In addition to high time efficiency, it is also able to leverage contextual information of pixels and can be optimized with a consistent loss. Experimental results on the widely-used datasets show that the proposed LAR-SR approach achieves superior performance on the visual quality and quantitative metrics compared with other generative models such as GAN, Flow, and is competitive with the mixture generative model.

1. Introduction

Recent years have witnessed great progress in deep learning based method for image super-resolution (SR) [5, 14, 33]. Most of the existing methods formulate image SR as a pixel-wise regression problem, which is optimized with a pixel-wise loss such as *L1* or *MSE*. As image super-resolution is inherently an ill-posed problem, when trained with many-to-one mapping between the high resolution images and the low resolution images, the regression-based models, with the per-pixel loss design, tend to adopt the average of all possible HR images, thus suffering from blurry and unreal SR images. See Figure 1(a) for an illustration.

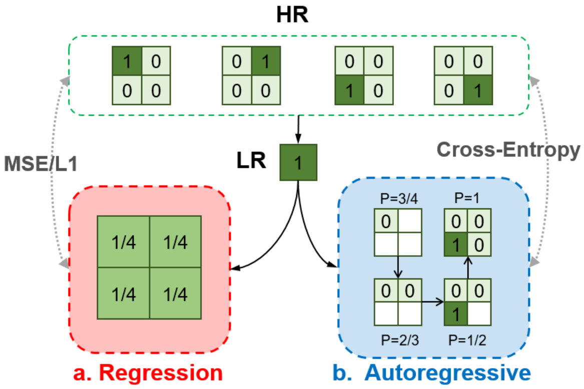

Figure 1. The purpose of regression-based methods (a) is to minimize the pixel-wise loss, i.e., *MSE-loss* or *L1-loss* between the ground truth and the output, which results in blur images lacking of details. Our algorithm is based on the autoregresssive method (b), which considers the relation between adjacent pixels. According to the HR datasets, the first pixel has a probability of 3/4 of 0 and a probability of 1/4 of 1. Once we sample the first pixel as 0, the second pixel has a posterior probability of 2/3 of 0 and a posterior probability of 1/3 of 1 et al.

To generate more realistic images, Generative Adversarial Network (GAN) [7]-based models and Flow [4]-based models have recently been introduced into image super-resolution. Both types of these methods generate all the pixels in parallel where the correlation between pixels is implicitly embedded in the latent space. During the training procedure, the discriminator or the invertable network maps the normal distribution to the joint distribution of the pixels. Thus GAN-based and Flow-based models can generate high-fidelity details compared with traditional regression-based models. However, GAN-based models pose the challenge of joint optimization, while Flow-based models are limited by the specific invertable network.

As another powerful generative model, autoregressive model has recently been explored in image synthesis tasks [6], which expressively learn relationships among its input. Because Taming transformer [6] is designed for general im-

[*] Equal contribution(co-first authors).
[†] Corresponding author.

age synthesis where global understanding of the input is acquired by modeling long-range relationship through sequential sampling, one great challenge of such expressive modeling is its computationally infeasibility for long sequences, especially for high-resolution images. Although patch-wise processing in a sliding-window manner has been adopted for speeding, its computation cost is still based on the size of the images and unacceptable especially for the super-resolution beyond High Definition (HD) resolution.

Focusing on the super-resolution task, with the LR input, we can obtain a coarse SR result by a simple regression model, which has already included the main content and semantic structure. And thus we only need the expressive model to generate the additional texture details or high frequency components of images, which can be modelled locally (experiments also verify this hypothesis). Therefore, in this paper we propose to take advantage of both regression and autoregressive models. With a coarse SR achieved by a regression model, we can efficiently and effectively generate texture details by a local autoregressive model with a learned texture codebook by our proposed LAR-SR model.

Specifically, for textural details generation, we partition the image into non-overlapping patches, and pixels in all the patches are sampled in parallel by the local autoregressive module as shown in Figure 2. Thus the time consumption is significantly reduced because of the parallelism. What's more, inspired by [20] that the AR-based model benefits from the quantization compression for the data space, a novel texture codebook based on Vector Quantized Variational AutoEncoder (VQVAE) [20] is adopted to learn and discretize the textural detail. Since this texture codebook is only for generating texture details, it is much easier to learn. Leveraging the texture codebook learned from the VQVAE and coarse-SR from the regression model, the proposed LAR-SR method can generate texture details efficiently in a patch-wise and parallel mode.

Our main contributions can be summarized as follows:

- We propose a novel local autoregressive super-resolution framework by taking advantage of both regression and autoregressive models, which can generate SR images with high-fidelity details but also high computation efficiency. To the best of our knowledge, it is the first AR-based framework designed for the super-resolution task.

- A novel local autoregressive (LAR) module is proposed to efficiently generate texture details in patch-wise and parallel mode, through a learned texture codebook from VQVAE and a coarse-SR from a regression model.

- We construct extensive experiments for two super resolution tasks: general super resolution and face super resolution. Objective quality metrics and visual results on three popular datasets (DIV2k [1], celebA [15] and FFHQ [10]) show that LAR-SR can yield state-of-the-art results compared with baseline approaches.

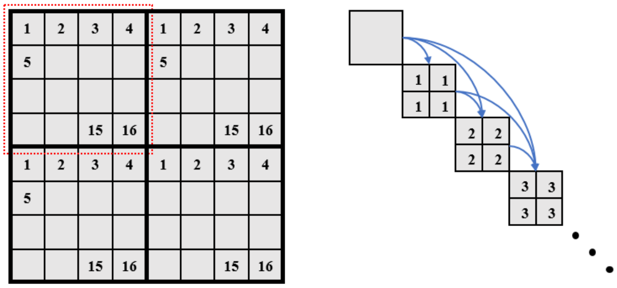

Figure 2. Example of our local autoregressive method, with a patch size of 4×4. All the pixels are labeled by their locations in each patch. The same labeled pixels are sampled in parallel. Thus the sampling time complexity only depends on the size of the patch.

2. Related Work

2.1. Perceptual-Oriented Super-Resolution

Regression-based methods such as RCAN [33], RRDB [28] and EDSR [14] aim to pursue pixel level restoration, which suffer from the blurry SR images. As shown in part (a) of Figure 1, multiple high resolution images corresponding to the same low resolution image after degradation and reducing the pixel-wise loss results in unreal patterns. Thus Perceptual-oriented methods include the GAN-based and Flow-based models are proposed to generate more realistic SR images. The GAN-based models are the most popular generative models for perceptual-oriented super-resolution. SRGAN [12] combine the adversarial loss with perceptual loss [9] to improve the visual quality. SFTGAN [27] propose a novel spatial feature transform to incorporate the semantic priors to generate rich and realistic textures. ESRGAN [28] enhances the original SRGAN by the modification of the architecture and the loss function. Moreover, recent works introduce the normalization flow [4] into super-resolution tasks. The flow-based SR models SRflow [16] introduce the normalization flow into super-resolution tasks. Then HCflow [13] adopts the multi-layer structure based on SRFlow to achieve a better performance. Moreover, HCFlow++ [13] combines the Flow-based and GAN-based models to generate more realistic SR images.

Both GAN-based and Flow-based models implicitly model the correlation between pixels and have their own limitations as we mentioned. Thus we adapt the autoregression model to the super-resolution to explicitly model the pixel-level correlation with a flexible network structure, which can be optimized by a single consistent loss.

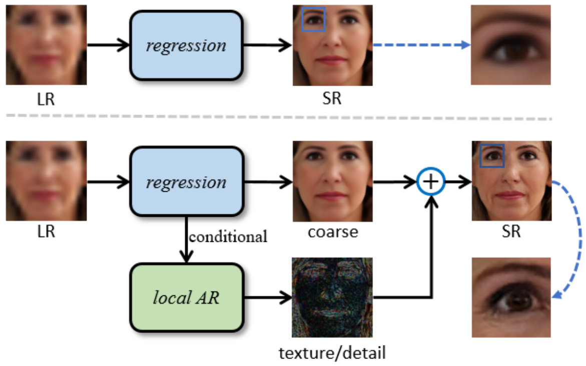

Figure 3. Data flow of the regression-based method and LAR-SR. LAR-SR divides the image into structural components and textural details. A regression-based module is used to restore the basic structure, and the details are then sampled by a local autoregressive (AR) module based on the recovered structure.

2.2. Autoregressive Models

Autoregressive models are common probabilistic models that fully factorize the probability density function with powerful generation and stable training procedures. PixelRNN [24] and PixelCNN [19] are the first proposed AR-based models for image generation. To improve the PixelCNN model, PixelCNN++ [21] replaces the full 256-way softmax with a logistic mixture likelihood to estimate the distribution of the pixels. On the other hand, VQVAE [20] is proposed to learn a discrete representation, which is utilized to build a powerful generative model with PixelCNN. Besides, Taming transformer [6] combined VQVAE with adversarial loss and transformer [25] for image generation.

Autoregressive model has demonstrated the promising results in image generation. But it is still inapplicable in low level tasks due to its severe time complexity. Our proposed LAR-SR adapts autoregressive model to the field of super-resolution. LAR-SR generates a discrete representation first only for the components of textural details in natural images, which are then be leveraged by a novel local autoregressive module (shown in Figure 2). Thus the time consumption is highly reduced while the model can generate a high-fidelity SR image.

3. Methodology

Our LAR-SR model follows a two-stage approach: in Stage 1, a textural VQVAE (tex-VQVAE) extracts and encodes the components of textural details in images into a discrete latent space. A local autoregressive model is proposed in Stage 2 based on the latent representation obtained from Stage 1. As the data flow shown in Figure 3, the structural components of the output images are generated from a regression network, i.e., a coarse SR module. The optimization for the both stages is individual, i.e., the learned tex-VQVAE is fixed in Stage 2. See appendix for more details about network structure.

3.1. Stage 1:textural VQVAE

VQVAE [20] designs a discrete learnable codebook with all the components of the images into the latent representations for image generation. For super-resolution, as the regression-based method can well restore the structural components, textural VQVAE (tex-VQVAE) is proposed to focus on the textural details in natural images. The tex-VQVAE includes an encoder $E(\cdot)$ and a decoder $D(\cdot)$. Given an input HR x, its feature vector $y_{i,j}$ at each pixel-wise position (i,j) is obtained using the encoder $y = E(x)$. Then the feature vector $y_{i,j}$ is replaced by its nearest prototype vector in the texture codebook $\mathbf{z} = \{z_k \mid k \in 1...K\}$ to obtain its quantized representation $\hat{y}_{i,j}$. This mapping is determined according to the distance between the feature vectors $y_{i,j}$ and z_k, as described in Equation (1).

$$\hat{y}_{i,j} = z_l, \text{where } l = \operatorname*{argmin}_k \|y_{i,j} - z_k\|, \tag{1}$$

we denote the mapped indices as $\mathbf{I}$, the element in $\mathbf{I}$, i.e., $\mathbf{I}(m,n)$, can be obtained by $\mathbf{I}(m,n) = \text{argmin}_k \|y_{m,n} - z_k\|$ as shown in Stage 1 in Figure 4. The codebook maps the indices back to the corresponding vectors to get $\hat{y}$. Unlike VQVAE, an extra input from a coarse SR module $C(\cdot)$ is added to the decoder in the proposed model, i.e., $\hat{x} = D(\hat{y}, x_c)$, where $x_c = C(x \downarrow)$ is the coarse SR image, to restore the structural components by regression-based method. Note that ↓ represents the degradation process (e.g., Bicubic in this paper) for generating the training pairs. The codebook can therefore focus more on the textural components to restore the image thanks to the extra input. Meanwhile, the goal of tex-VQVAE is to close up the distance between $\hat{x}$ and the input x. The *commitment loss* and *codebook loss* are also applied to solve the non-derivable operations in the encoder [20]. The total objective can be formulated as Equation (2):

$$\begin{aligned}\mathcal{L}(x,\hat{x},x_c) = \|x-\hat{x}\| + \|sg[y]-\hat{y}\| \\ + \beta\|sg[\hat{y}]-y\| + \|x_c - x\|,\end{aligned} \tag{2}$$

where the operator sg refers to a stop-gradient operation and β is a hyper-parameter. The modified tex-VQVAE can be used to extract and quantize the textural components.

In summary, the feature map extracted by the encoder network is downsampled by a factor of two and then quantized by the codebook. The quantized representation maintains the components of textural details in our design. Meanwhile, the low resolution image is input to a coarse super-resolution module. Both the quantized representations and the coarse SR image are the inputs of the decoder, which consists of a few residual blocks and a transposed convolution layer. Since the decoder receives the structural and textural components from the inputs, it can restore the whole image accurately.

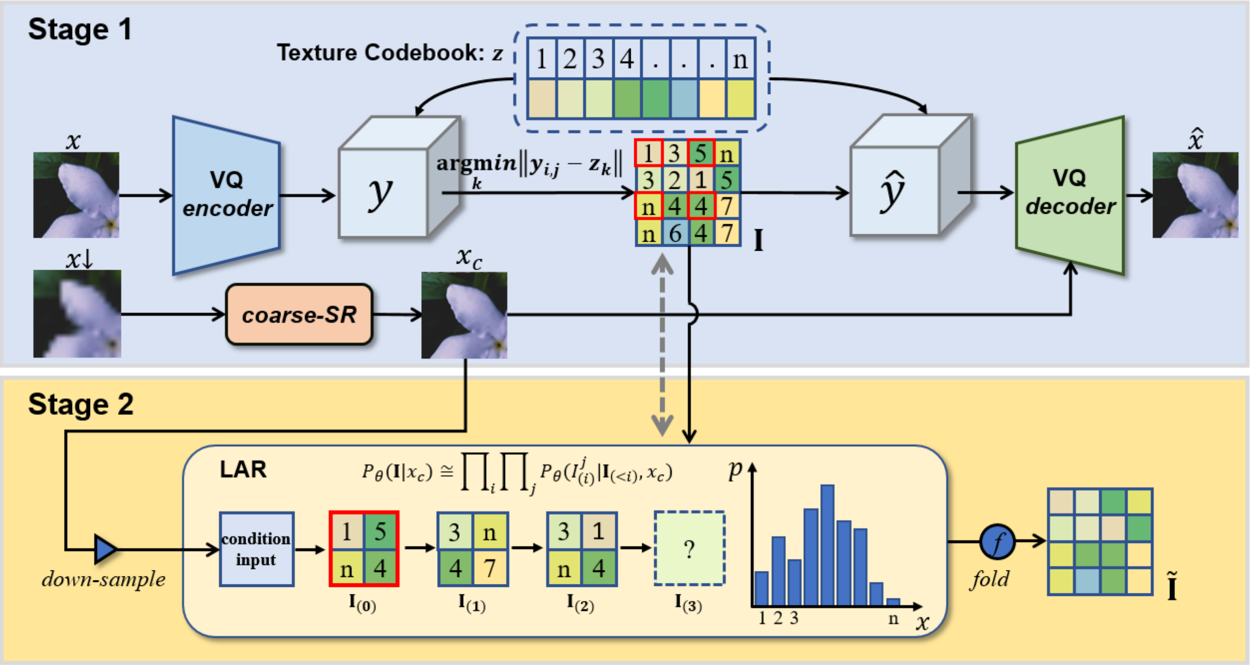

Figure 4. The overall pipeline of the proposed LAR-SR. The architecture follows a two-stage approach: in Stage 1, a customed textural VQVAE (tex-VQVAE) extracts and discretizes the textural components of the HR images in the training dataset. Then in Stage 2, a local autoregressive model is trained over the textural components' embedding to fit a posterior probability.

3.2. Stage 2: Local Autoregressive Model

The goal of Stage 2 is to learn a posterior probability distribution with a *cross-entropy loss* over the textural indices $\mathbf{I}$ conditioned on the corresponding coarse SR image x_c, which is practically better than the origin LR image. The main idea of the traditional autoregressive model is to convert a joint probability distribution into a product of conditional distributions:

$$\mathcal{P}_\theta(\mathbf{I}) = \prod_{i=1}^{n} \mathcal{P}_\theta(I_i \mid \mathbf{I}_{<i}), \tag{3}$$

where I_i is the i-th element in $\mathbf{I}$ from the top-left pixel to the bottom-right pixel in sequence, and $\mathbf{I}_{<i} = \{I_j \mid j < i\}$. Thus, the traditional autoregressive model can be seen as a global autoregressive model, which results in enormous computational complexity. More specially, to generate an image of size $H * W$, the model needs to propagate forward $H * W$ times. To solve this problem, in this paper a local autoregressive model is proposed to relieve such huge time consumption. The diagram of the local autoregressive model is shown in Stage 2 of Figure 4. Based on the fact that the textural components of the images are almost locally correlated given the structural components, the images are divided into non-overlapping patches. In the local autoregressive model, image patches can be sampled in parallel. However, these patches are not completely independent due to the boundary of adjacent patches and the consistency of generated data. To achieve this, the pixels in each patch are sequentially labelled in the same way. As an example in Figure 2, with the patches of size $4 * 4$, all the pixels are labeled from 1 to 16 by their location and sampled in sequence, which can be formulated as follows:

$$\mathcal{P}_\theta(\mathbf{I}|x_c) = \prod_{i=1}^{k} \mathcal{P}_\theta(\mathbf{I}_{(i)}|\mathbf{I}_{(<i)}, x_c) \simeq \prod_{i=1}^{k} \prod_{j} \mathcal{P}_\theta(I^j_{(i)}|\,\mathbf{I}_{(<i)}, x_c), \tag{4}$$

where k is the number of the pixels in a patch and x_c refers to the coarse SR image, $\mathbf{I}_{(i)}$ is the set of the pixels I_j with same location i, $\mathbf{I}_{(<i)} = \{\mathbf{I}_{(j)} \mid j < i\}$ refers to the set of all the $\mathbf{I}_{(j)}$ with j is smaller than i, and $I^j_{(i)}$ is the j-th element of $\mathbf{I}_{(i)}$. It's worth noting that $\mathbf{I}_{(<i)}$ is different from $\mathbf{I}_{<i}$. Because the textural components are nearly locally related, each two elements in $\mathbf{I}_{(i)}$ are almost independent. Thus, all the elements $I^j_{(i)}$ in $\mathbf{I}_{(i)}$ can be sampled in parallel. In this way, the time complexity of the sampling procedure of the LAR only depends on the size of the patch. Thus, it is significantly lower than that of the global autoregressive model. *Cross-entropy loss* is used to optimize the estimation for the distribution $\mathcal{P}_\theta(I^j_{(i)} \mid \mathbf{I}_{(<i)}, x_c)$.

LAR module consists of three parts: two encoders to input the coarse SR image x_c and the textural indices $\mathbf{I}$, l LAR-layers and an output module $\mathcal{G}$. Given the coarse SR image

x_c and the textural indices $\mathbf{I}$, the initial input for the LAR-layers can be calculated by

$$f = enc_1(x_c) \oplus enc_2(\mathbf{I}), \tag{5}$$

$$\{f_1, f_2 \dots, f_{s\times s}\} = unfold(f, s), \tag{6}$$

where $\oplus$ means to concatenate on the feature dimension, s means the local patch size (as shown in Figure 2, the patch size is 4). enc_1 consists of 3×3 convolutional layers, while enc_2 only consists of 1×1 convolutional layers to avoid seeing future pixels during training. Denote $\{f_i^{(j)} \,|\, i = 1, 2 \dots s \times s\}$ as the output of the j-th LAR-layer and the input of the $(j+1)$-th layer. In order to unify the next calculation process at different LAR-layers, $f_i^{(0)}$ is initialized as f_{i-1} except $f_1^{(0)}$, which is initialized as $enc_1(x_c) \downarrow$, where $\downarrow$ means a downsampling process. Then the LAR-layer can be fomulated by:

$$f_i^{(j+1)} = C_{3\times3}(C_{1\times1}(f_1^{(j)} \oplus f_2^{(j)} \oplus \dots \oplus f_{i-1}^{(j)}))), \tag{7}$$

The output of the last LAR-layer is folded and input to the output convolutional layer:

$$\mathcal{P}_\theta(\mathbf{I}|x_c) = softmax(\mathcal{G}(fold(f_1^{(l)}, f_2^{(l)} \dots, f_{s\times s}^{(l)}))), \tag{8}$$

where l is the number of the LAR-layers and $\mathcal{G}$ is the output convolutional module. During the inference procedure, $\tilde{\mathbf{I}}$ is sampled from $\mathcal{P}_\theta(\mathbf{I} \,|\, x_c)$. Then the SR image is generated by $\tilde{x} = D(\tilde{y}, x_c)$, where $\tilde{y}$ is mapped back by $\tilde{\mathbf{I}}$ through the texture codebook obtained in Stage 1, and D is the decoder of the tex-VQVAE which is also from Stage 1.

4. Experiments

We conduct our experiments on the widely-used super-resolution dataset DIV2K [1](4×) following [13,16]. Moreover, to further demonstrate the effectiveness of the proposed LAR-SR, we also conduct the additional experiments on the face image super-resolution task using celebA [15] dataset (8×) and FFHQ [10] dataset. Recent works such as [26, 30] have made significant progress in face super-resolution (FSR). However, these methods are based on a pretrained StyleGAN [10] for face images which consumes huge amounts of computing resources and is also difficult to be adapted to other types of images. Thus, for FSR task, we compare LAR-SR with more general approaches.

4.1. Experiments Setting

General Super-Resolution: For general image super-resolution (4×), the both modules in Stage 1 and Stage 2 are trained and validated on the training dataset of DIV2K [1] and Flicker2K [22], and finally tested on the validation dataset of DIV2K. RRDB [28] is used as the coarse SR module. The size of the dictionary in Stage 1 is 1024. The number of LAR modules in Stage 2 is set as 12. The data augmentation strategy during training includes random flipping and random crop (160×160 patch size). The batch size is set to 32. Adam optimizer [11] is used with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $weightdecay = 1 \times 10^{-7}$. The learning rate for the Stage 1 is set to 1×10^{-4} during the fully 30 epochs. And the learning rate in Stage 2 is initialized as 1×10^{-4} at the first 40 epochs, and set to 1×10^{-5} at the last 20 epochs. The tex-VQVAE in Stage 1 is trained first and is fixed during the training procedure of Stage 2.

Face Super-Resolution Following [3, 18], we evaluate LAR-SR on two datasets: CelebA [15] and FFHQ [10]. Both datasets are popularly used for evaluating FSR performance. For each dataset, 90% of the images are used for training and the rest are used for testing. The HR face images are cropped and resized to the 128×128 resolution. We use random flipping method as the data augmentation and use SPARNet [2] as the coarse SR module. The size of the dictionary is set as 128. The number of LAR layers is set as 12. Other training configurations are same as the configurations in general super-resolution task.

4.2. Metrics For valuation

For evaluating the performance of models, LPIPS [31] metric, which is proven to be correlated well with human visual perception, is used as the metric for the evaluation of the image perceptual quality. Peak Signal to Noise Ratio (PSNR) and Structural Similarity (SSIM) [29] are also reported in order to compare our method with other existing methods objectively, although they are known to be not correlated well with the image quality for super-resolution [8, 17, 23].

4.3. Result and Analysis

General Super-Resolution: We compare the proposed LAR-SR method with various state-of-the-art methods. EDSR [14] and RRDB [28] are the PSNR-oriented models. GAN-based models include ESRGAN [28] and RankSR-GAN [32]. Flow-based models include SRFlow [16] and HCFlow [13]. The quantitative results are shown in Table 1. We further visualize the SR images in Figure 5. From Table 1 we can see that LAR-SR method outperforms the baseline on the LPIPS metric and yields competitive PSNR and SSIM performance comparing to the GAN and flow-based approaches. Compared with the mixture generative model HCFlow++, LAR-SR gives a similar LPIPS and higher PSNR and SSIM. Figure 5 also demonstrates that LAR-SR has competitive detail-generative ability with the fidelity to the corresponding LR images.

Face Super-Resolution: We further test LAR-SR on general face super-resolution tasks. As we mentioned above, the state-of-the-art methods [26, 30] are based on GAN in-

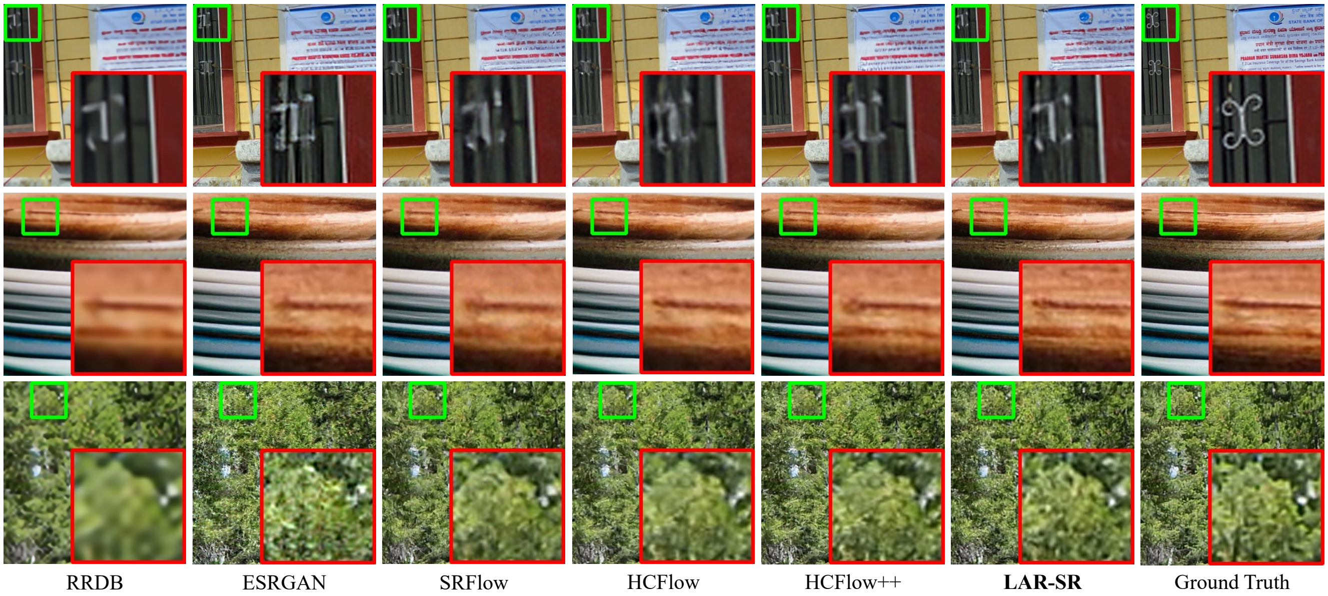

Figure 5. Visual results of different methods for general image SR (×4) [1].

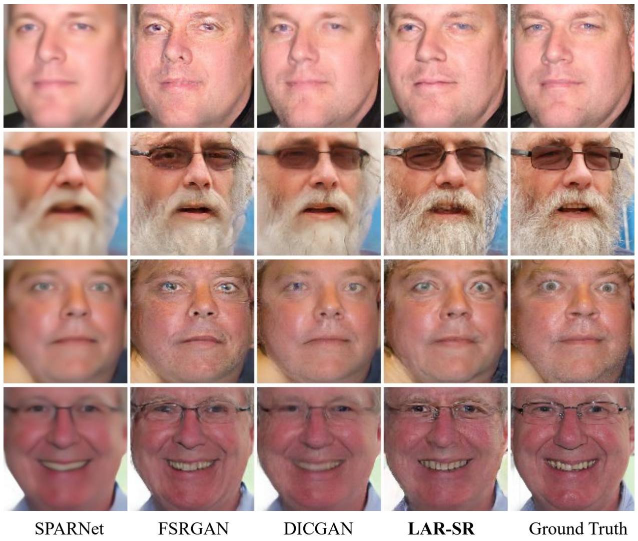

Figure 6. Visual results of different methods for face image SR (×8) [10].

version which requires a pretrained GAN model of face images, thus they are difficult to be adapted into other low level tasks. As an additional experiments, we only compare LAR-SR with several regular regression and generative methods. For GAN-based methods, instead of RankSRGAN and ESRGAN for general super-resolution, FSRGAN [3] and DICGAN [18] are used as the baselines. And the regression based methods include SPARNet [2] and DICNet [18]. Note that all the models are retrained due to various preprocessing in different FSR models. It can be seen that in Table 2 and Figure 6, LAR-SR outperforms all the benchmarks on LPIPS metric with competitive PSNR and SSIM. The visualization results further illustrate the efficiency of LAR-SR, where it shows that LAR-SR method generates more realistic details with less artifacts.

Method		PSNR↑	SSIM↑	LPIPS↓
	Bicubic	26.70	0.77	0.409
Reg.-based	EDSR [14]	28.98	0.83	0.270
	RRDB [28]	29.44	0.84	0.253
GAN-based	ESRGAN [28]	26.22	0.75	0.124
	RankSRGAN [32]	26.55	0.75	0.128
Flow-based	SRFlow [16]	27.09	0.76	0.121
	HCFlow [13]	27.02	0.76	0.124
Flow+GAN	HCFlow++ [13]	26.61	0.74	0.110
	LAR-SR	27.03	0.77	0.114

Table 1. Quantitative comparison for general image SR (×4) on DIV2K [1] validation set.

LAR-attn-SR: Although the time consumption for LAR-SR is massively reduced compared with the traditional AR-based models, it still costs more than ten seconds during the sampling procedure. We further propose a more lightweight, patch size independent LAR-attn-SR by adopting local mask attention, and the Formula 7 is changed to:

$$f_i^{(j+1)} = C_{3\times3}(\sum_{t=1}^{i} a_{it} f_t^{(j)}), \tag{9}$$

where $C_{3\times3}$ means 3×3 convolutional layer and the attention weight a_{it} can be calculated by:

$$e_{it} = g(f_i^{(j)}, f_t^{(j)}) = \boldsymbol{W}(f_i^{(j)} \oplus p_i)\boldsymbol{U}(f_t^{(j)} \oplus p_t), \tag{10}$$

$$a_{it} = \frac{exp(e_{it})}{\sum_{k=1}^{i} exp(e_{ik})}, \tag{11}$$

Method		PSNR↑	SSIM↑	LPIPS↓
CelebA				
Reg.-based	DICNet [18]	28.84	0.838	0.174
	SPARNet [2]	29.01	0.845	0.165
GAN-based	FSRGAN [3]	27.15	0.780	0.085
	DICGAN [18]	27.58	0.792	0.118
	LAR-SR	27.26	0.784	0.077
FFHQ				
Reg.-based	DICNet [18]	27.69	0.801	0.185
	SPARNet [2]	27.78	0.800	0.219
GAN-based	FSRGAN [3]	25.25	0.785	0.118
	DICGAN [18]	26.03	0.742	0.100
	LAR-SR	25.66	0.740	0.088

Table 2. Quantitative comparison for face image SR (×8) on CelebA and FFHQ test set. Note that PSNR is calculated on Y-channel following [2, 18]

Method	PSNR	SSIM	LPIPS	Time	SizeofModel
LAR-SR	27.03	0.77	0.114	14.7s	62.1M
LAR-attn-SR	27.23	0.79	0.118	7.8s	10.1M

Table 3. Comparison between LAR-SR and LAR-attn-SR. Tested on the validation set of DIV2K on a NVIDIA Tesla V100 GPU.

Patch size	PSNR	SSIM	LPIPS	Time/s
1×1	27.21	0.775	0.163	1.4
2×2	26.98	0.774	0.120	2.3
3×3	27.08	0.775	0.116	6.9
4×4	27.04	0.776	0.114	14.7

Table 4. Quantitative comparison for different patch sizes on DIV2K validation set [1]. Tested on a NVIDIA Tesla V100 GPU.

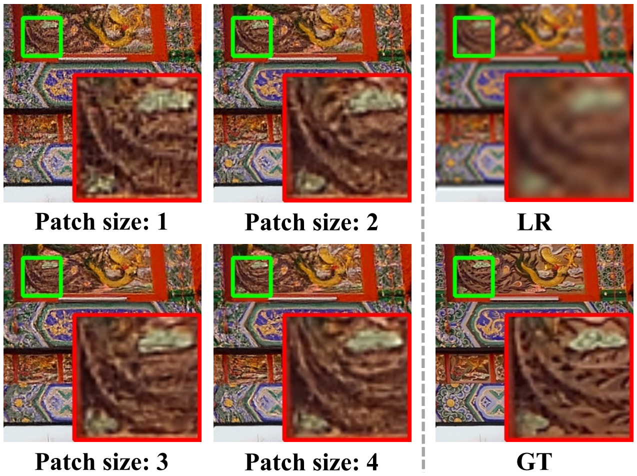

Figure 7. Visual results of different patch sizes in LAR-SR.

where $\boldsymbol{W}$ and $\boldsymbol{U}$ are linear projection matrices and p_i, p_t are the position encoding. The quantitative comparison between LAR-SR and LAR-attn-SR with patch size 4 is shown in Table 3. The model size is greatly reduced, and the sampling time is reduced as well, but still with comparable performance.

Study of the patch size: LAR-SR is based on the assumption that the the textural components are local-related, and uses patch-level local autoregression for the textural components to reduce the huge time consumption in traditional autoregression method. Thus it's important to validate the patch size trade off between the speed and the performance. In this section, we conduct a study to further evaluate the effects of the patch size in LAR-SR. Specifically, the patch size is set from 1 to 4 in the experiments while other settings remain unchanged. The quantitative results are shown in Table 4. There are several observations from Table 4. First, The sampling time consumption is significantly reduced by LAR-SR, which makes the AR-based method promising in low-level tasks. Second, as the patch size increases, the quantitative and visualization results are also improved. Third, as the patch size increases, the gain brought by increasing the patch size decreases rapidly. This validates the locality of the textural components. The visualization results for different patch sizes are shown in Figure 7, which shows a similar trend as that in Table 4.

Comparison with global autoregressive model: It is necessary to compare LAR-SR with traditional autoregressive models to validate the advantages of LAR-SR. Because there is no existing AR-based methods which are specially proposed for super-resolution tasks, we construct and train a baseline VQVAE+pixelCNN method based on a VQVAE [20] and a conditional pixelCNN [19]. This VQVAE+pixelCNN method represents a ”traditional” AR-based method. First, we train the VQVAE to restore the HR images and then a conditional pixelCNN is used to estimate the posterior probability distribution conditional on the LR images. Due to the huge time consumption for traditional AR-based models, we only compare LAR-SR with the baseline VQVAE+pixelCNN method on CelebA [15] dataset. The quantitative results are shown in Table 5. It shows that LAR-SR can not only yield consistent performance gains over VQVAE+pixelCNN on all the quantitative metrics, but also process the images with a significantly lower time consumption. More specifically, our proposed method reduces the time consumption to about 0.52% of the global autoregressive method for images of resolution 128 × 128. It is worth noting that the time complexity of our proposed local autoregressive method is mainly based on the chosen patch size, which means that when the size of input images increases our method shows more advantages compared with global autoregressive method in the time consumption.

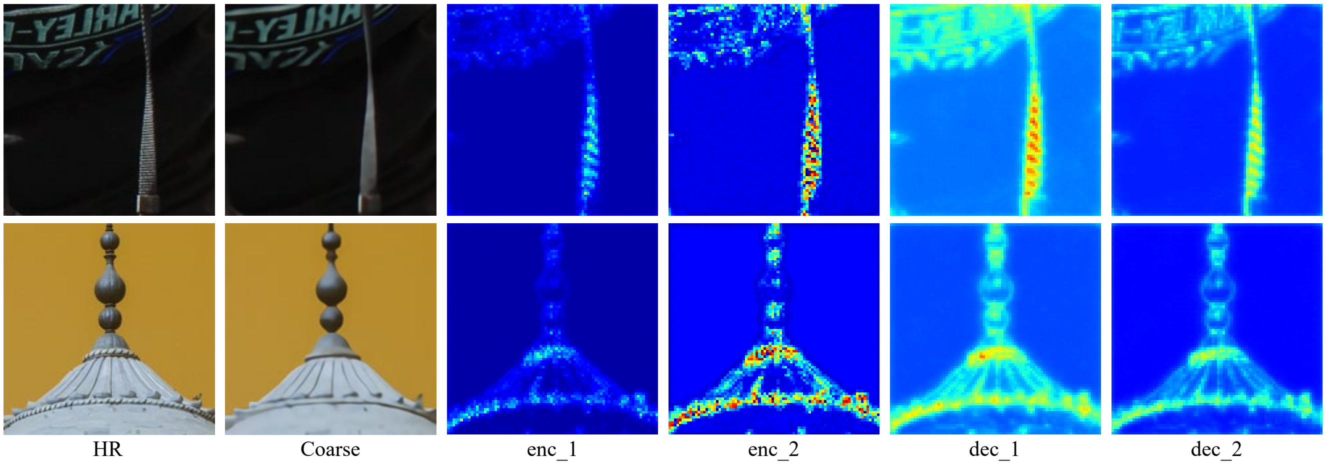

Figure 8. Visualization of the heatmap for the encoder and decoder in tex-VQVAE.

	Method	Time	PSNR	SSIM	LPIPS
AR	VQVAE+pixelCNN	230.7s	26.53	0.752	0.104
LAR	**LAR-SR**	1.2s	27.26	0.784	0.077

Table 5. The quantitative comparison between LAR-SR and the AR-based baseline.The time duration is measured with a NVIDIA RTX 1080Ti GPU.

Analyze of the texture codebook: We conduct another study to better investigate the effect of the text codebook and tex-VQVAE, for which we visualize the heatmaps for different layers in the encoder and the decoder of tex-VQVAE. The results are shown in Figure 8. As expected, it shows that tex-VQVAE indeed pays more attention to the textual information, especially the image components that are not well restored in the coarse SR image.

Ablation study: We further conduct ablation studies to measure the effects of the restored structural components in both stages. The structural component, i.e., x_c is restored by the regression-based coarse SR module. In Stage 1, x_c is used to construct the tex-VQVAE, while in Stage 2, the textural component $\tilde{\mathbf{I}}$ is sampled conditioned on x_c. In the first ablation study, structural branch is removed in stage one and replaced by the origin LR image in stage two and the model is named as LAR-vanilla. On the other hand, to compare the difference between VQVAE [20] and tex-VQVAE, we remove the structural input, i.e., x_c for the decoder in Stage 1, which makes the modified tex-VQVAE similar to the original VQVAE, and the model is denoted as LAR-full. Quantitative and visualization comparison are constructed between these two methods and LAR-SR. The quantitative results are shown in Table 6. As expected, the structural component is important in both stages. The comparison between LAR-vanilla and LAR-full illustrates that the final SR images rely on the restoration of the basic structure, while the comparison between LAR-full and LAR-SR shows that the textural-structural components separation brings less artifacts. Moreover, both the ablation studies show that the visual quality, i.e., LPIPS metric, gains more from the split of structure and texture in Stage 1.

Method	Stage 1	Stage 2	PSNR/SSIM/LPIPS
LAR-vanilla	✗	✗	26.31/0.75/0.185
LAR-full	✗	✓	26.56/0.76/0.164
LAR-SR	✓	✓	27.03/0.77/0.114

Table 6. Quantitative comparison of different models in ablation study.

5. Conclusion

In this paper, we propose a novel approach called LAR-SR for super-resolution task based on a tex-VQVAE and a local autoregressive module. To the best of our knowledge, it is the first work to adapt AR-based models into super-resolution. The experiments demonstrate that our proposed approach can yield state-of-the-art performance when compared with previous super-resolution models. Besides, we explore the implementation of LAR module and propose LAR-attn module to further improve the computation efficiency. In addition, the remarkable reduction of time complexity resulted from the localized autoregressive operation makes the proposed LAR-SR model much more applicable for more low-level tasks.

Acknowledgement

This work was supported in part by Chinese National Key R&D Program (2019YFB1804304), State Key Laboratory of UHD Video and Audio Production and Presentation, BirenTech Research, Shanghai Key Laboratory of Digital Media Processing and Transmissions(STCSM 18DZ2270700) and 111 plan(BP0719010).

References

[1] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017. 2, 5, 6, 7

[2] C. Chen, D. Gong, H. Wang, Z. Li, and Kyk Wong. Learning spatial attention for face super-resolution. *IEEE Transactions on Image Processing*, 30:1219–1231, 2021. 5, 6, 7

[3] Y. Chen, Y. Tai, X. Liu, C. Shen, and J. Yang. Fsrnet: End-to-end learning face super-resolution with facial priors. 2017. 5, 6, 7

[4] L. Dinh, D. Krueger, and Y. Bengio. Nice: Non-linear independent components estimation. *Computer Science*, 2014. 1, 2

[5] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Trans Pattern Anal Mach Intell*, 38(2):295–307, 2016. 1

[6] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. 2020. 1, 3

[7] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3:2672–2680, 2014. 1

[8] Andrey Ignatov, Radu Timofte, Thang Van Vu, Tung Minh Luu, Trung X Pham, Cao Van Nguyen, Yongwoo Kim, Jae-Seok Choi, Munchurl Kim, Jie Huang, et al. Pirm challenge on perceptual image enhancement on smartphones: Report. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 5

[9] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. 2

[10] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 2, 5, 6

[11] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[12] C. Ledig, L. Theis, F Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, and Z. Wang. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE Computer Society*, 2016. 2

[13] Jingyun Liang, Andreas Lugmayr, Kai Zhang, Martin Danelljan, Luc Van Gool, and Radu Timofte. Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4076–4085, 2021. 2, 5, 6

[14] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017. 1, 2, 5, 6

[15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 2, 5, 7

[16] A. Lugmayr, M. Danelljan, L Van Gool, and R. Timofte. Srflow: Learning the super-resolution space with normalizing flow. 2020. 2, 5, 6

[17] Andreas Lugmayr, Martin Danelljan, and Radu Timofte. Unsupervised learning for real-world super-resolution. in 2019 ieee. In *CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3408–3416, 2019. 5

[18] C. Ma, Z. Jiang, Y. Rao, J. Lu, and J. Zhou. Deep face super-resolution with iterative collaboration between attentive recovery and landmark estimation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 5, 6, 7

[19] Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. *arXiv preprint arXiv:1606.05328*, 2016. 3, 7

[20] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017. 2, 3, 7, 8

[21] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017. 3

[22] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017. 5

[23] Radu Timofte, Shuhang Gu, Jiqing Wu, and Luc Van Gool. Ntire 2018 challenge on single image super-resolution: Methods and results. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 852–863, 2018. 5

[24] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756. PMLR, 2016. 3

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3

[26] Xintao Wang, Yu Li, Honglun Zhang, and Ying Shan. Towards real-world blind face restoration with generative facial prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9168–9178, 2021. 5

[27] X. Wang, K. Yu, C. Dong, and C. C. Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. *IEEE*, 2018. 2

[28] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, C. C. Loy, Y. Qiao, and X. Tang. Esrgan: Enhanced super-resolution generative adversarial networks. *Springer, Cham*, 2018. 2, 5, 6

[29] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to

structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[30] Tao Yang, Peiran Ren, Xuansong Xie, and Lei Zhang. Gan prior embedded network for blind face restoration in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 672–681, 2021. 5

[31] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

[32] W. Zhang, Y. Liu, C. Dong, and Y. Qiao. Ranksrgan: Generative adversarial networks with ranker for image super-resolution. 2019. 5, 6

[33] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. Image super-resolution using very deep residual channel attention networks. 2018. 1, 2