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

E	*encoder of tex-VQVAE*
D	*decoder of tex-VQVAE*
C	*coarse SR module in encoder*
x	*HR image*
x_c	*coarse SR image (structural components)*
$\mathbf{z}$	*texture codebook*
$\mathbf{I}$	*codebook indices for textural components*

Table 1. Notations used in Algorithm 1 and 2

A. Algorithm Flow

Our proposed local autoregressive model for image super resolution (LAR-SR) follows a two stage approach. In Stage 1, a customed VQVAE [7] called textural VQVAE is proposed to extract and discrete the textual details in HR images. In Stage 2 a local autoregressive module is used to model the posteriori probability of the textural components conditioned on the structural components. Note that modules in Stage 1 is supposed to be fixed in Stage 2. The algorithms for both stages are shown in Algorithm 1 and 2. The notations defined in the algorithms are shown in Table 1.

Algorithm 1 textural VQVAE training

```
Require: Functions $E, D, C, x$ (batch of training images)
    $x \downarrow \leftarrow x$
    $y \leftarrow E(x)$

    // Quantize with texture codebook $\mathbf{z}$
    $\hat{y} \leftarrow y$

    // Restore the structural components by coarse SR
    $x_c \leftarrow C(x \downarrow)$

    $\hat{x} \leftarrow D(\hat{y}, x_c)$

    // Optimization for tex-VQVAE
    $\mathcal{L}(x, \hat{x}, x_c) \leftarrow (x, \hat{x}, x_c, y, \hat{y})$

    $\theta(E, D, C) \leftarrow Update(\mathcal{L}(x, \hat{x}, x_c))$
```

Algorithm 2 LAR module training

```
Require: Functions LAR Module, $E, C, D$, $x$ (batch of training images)
    // Quantize with texture codebook $\mathbf{z}$
    $\mathbf{I} \leftarrow quantize(E(x))$

    $x \downarrow \leftarrow x$
    $x_c \leftarrow C(x \downarrow)$

    // Training LAR module with cross-entropy
    $\mathcal{P}$ =TrainingLARModule($\mathbf{I}, x_c$)

    *************Sampling Procedure**************
    $\tilde{\mathbf{I}} \sim \mathcal{P}(x_c)$
    // Map back by codebook $\mathbf{z}$
    $\tilde{y} \leftarrow \tilde{\mathbf{I}}$

    $\tilde{x} = D(\tilde{y}, x_c)$
```

B. LAR-layer

We have illustrated the forward propagation of LAR module, which mainly consists of LAR-layers. LAR-layer has been designed to avoid seeing future information. The data flow of LAR-layer is shown in Figure 1. To unify the calculation process in different LAR-layers, the last pixels in each patches are dropped and the condition inputs added to the first position of the autoregression. To reduce the size of parameters, we use a customed block named split conv module, which consists of a 1×1 convolutional layer and a 3×3 convolutional layer. The size of LAR-layer increases as the patch size increases, thus we propose a novel architecture, LAR-attn-layer, to lightweight the LAR module and therefore it can be used with a large patch size. Different structures of the LAR module are worth exploring in future.

C. Extend experiments

C.1. training stability

GAN-based methods for super resolution pose the challenge of joint optimization due to the mismatch between the pixel-wise and adversarial losses. The pixel-wise losses tends to generate blurry and unreal patterns, which is contradictory to adversarial loss. To validate that the proposed LAR-SR has better training stability than GAN-based methods, we show the comparison of training stability between LAR-SR and GAN-based methods during the training process in Figure 2. It demonstrates the metrics' change on a subset of the test set (e.g., 1/10 of the testset) with the training epochs for our and GAN-based models. As shown in Figure 2, the inherent instability of adversarial loss and the contradiction between adversarial loss and pixel-wise loss leads to the polylines' oscillation on the GAN-based models during training. By contrast, LAR-SR model is able to maintain a continuous decline and give a final convergence. Also, during training, the PSNR and the LPIPS are improved synchronously with the LAR-SR, but they fluctuate with the GAN-based models, which further validates the advantages of the LAR-SR model in terms of training stability comparing with GAN-based models.

C.2. Effect of coarse SR module

Another experiment is conducted to investigate the effect of the coarse SR module. Two different regression network structures and a pure BICUBIC upsampling method are used as the coarse SR module to train three models on celebA [4] dataset. The results of these three different methods are shown in Table 2. It can be seen that higher PSNR of the coarse SR module yields better LAR-SR model. We speculate that better coarse SR module can not only provide more accurate structural information, but also can make the codebook and the LAR module in stage2 focus more on the textural components.

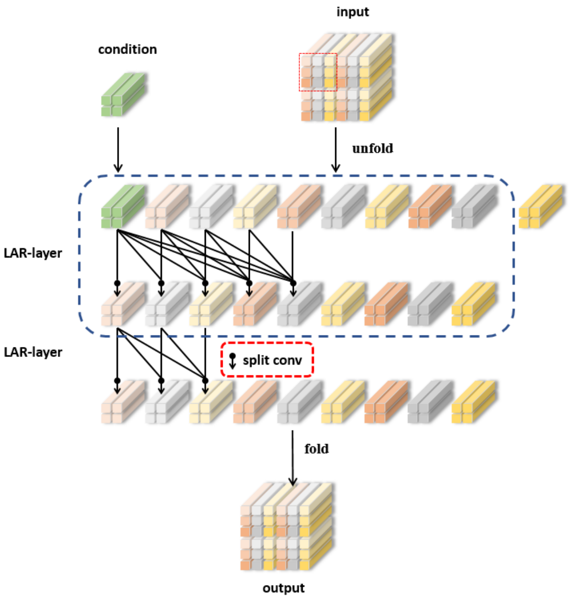

Figure 1. The data flow of LAR-layer during training. Split conv module is used to reduce the number of parameters. The condition is from the structural components.

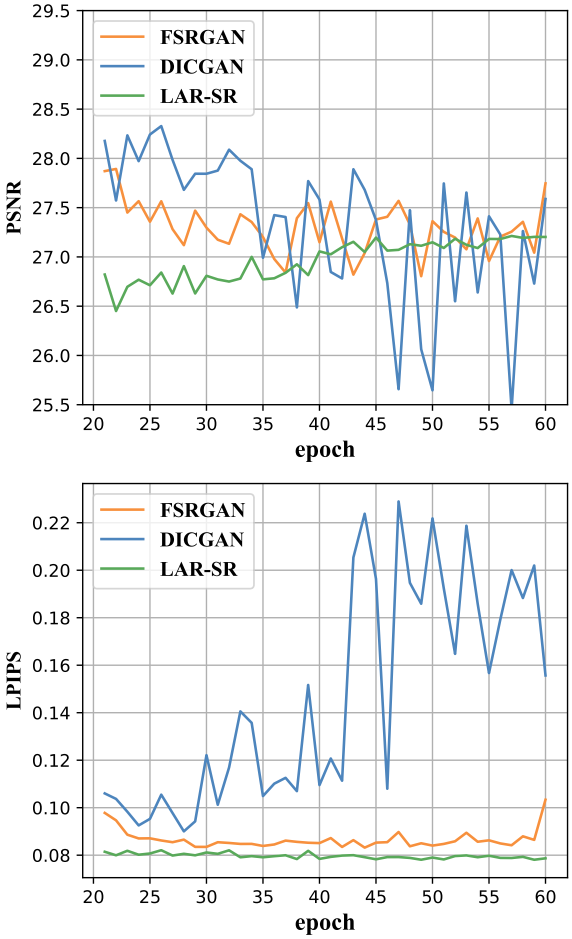

Figure 2. Comparison between LAR-SR and GAN-based methods [2, 5] during the training process, note that for the first 20 epochs the GAN-based methods only adopt pixel-wise loss to initialize the model.

coarse SR	PSNR	SSIM	LPIPS	PSNR*
BICUBIC	26.89	0.774	0.085	24.87
SRResnet [3]	27.25	0.785	0.082	28.51
SPARNet [1]	**27.26**	**0.784**	**0.077**	**29.01**

Table 2. Quantitative comparisons for LAR-SR with various coarse SR modules on CelebA-test. PSNR* refers to the PSNR metric of the coarse SR module

D. Network Structure

In this section we show the details of the network structure of LAR-SR. The network structure of Stage 1 is shown in Figure 3. The goal of Stage 1 is to restore the HR image. The input includes the HR image and the LR image. Because of the extra input of LR image and the coarse SR module, the codebook can pay more attention to the textual details. Note that the textural indices, i.e., the discrete latent codes for the codebook, is downscaled by the encoder and upscaled by the decoder. In Stage 2, as shown in Figure 4, the input includes the textural indices and the condition input, i.e., the structural components. Only 1×1 convolutional layers are used to avoid seeing future information. The details about LAR-layers are described in Section B.

E. Limitation & Future work

Although LAR-SR successfully adapts the AR-based model into low level tasks, there are still some limitations. The observed problems for LAR-SR and the potential solutions are analyzed in this section. LAR-SR sometimes generates unreal patterns randomly, especially for artificial landscapes. Such artifacts may be caused by the local autoregression module with the lack of the enough global information. The combination of AR-based and GAN-based model [6] can be promising to solve the problem. And the introduction of semantic information may also be helpful [8]. Moreover, although the local autoregression massively reduces the computation complexity of traditional AR-based models, there is still a gap on the processing speed between LAR-SR and regression-based methods. Thus, we have proposed LAR-attn-SR in this paper to further improve the computation efficiency. Better structures of LAR module should be explored in future.

References

[1] 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.

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

[3] 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.

[4] 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.

[5] 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.

[6] Alireza Makhzani and Brendan Frey. Pixelgan autoencoders. *arXiv preprint arXiv:1706.00531*, 2017.

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

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

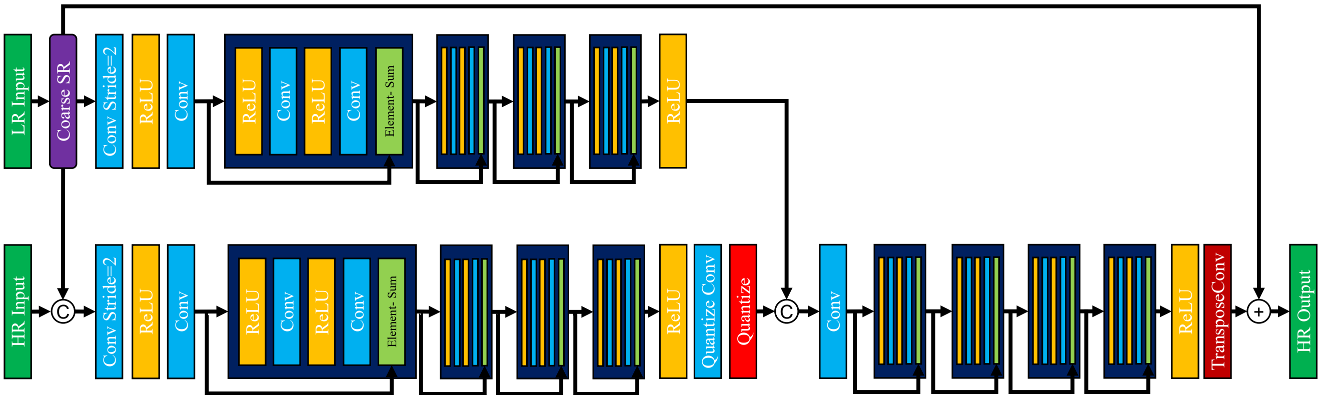

Figure 3. Architecture of Stage 1 in LAR-SR. The input includes both the HR image and the LR image and the output is to restore the HR image. Thanks to the extra structural input, i.e., the coarse SR image for the decoder, the codebook is able to extract the textural details in the HR image.

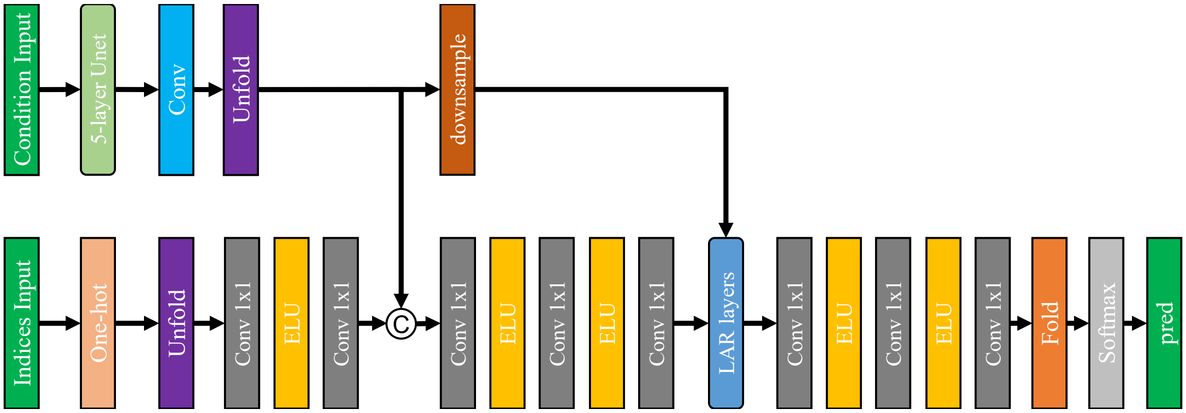

Figure 4. Architecture of Stage 2 in LAR-SR. The input includes the indices input and the the condition input. The target of Stage 2 is to model the posterior probability of the textural indices conditioned on the structural components, i.e., the coarse SR image.

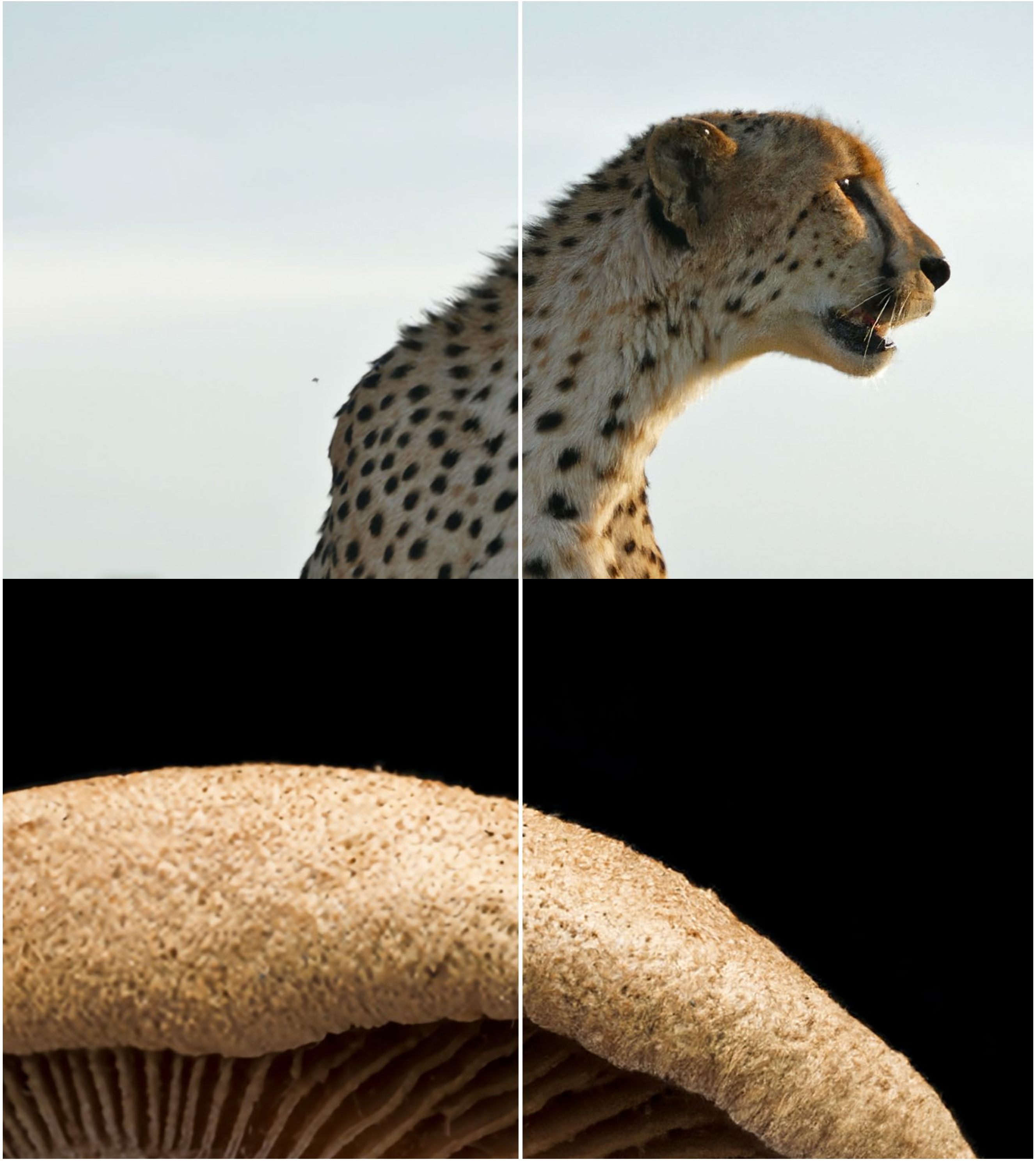

Figure 5. Additional visual result of super-resolution images. The left parts of the images are generated by a Regression-based method, the right parts of the images are generated by the proposed LAR-SR method.

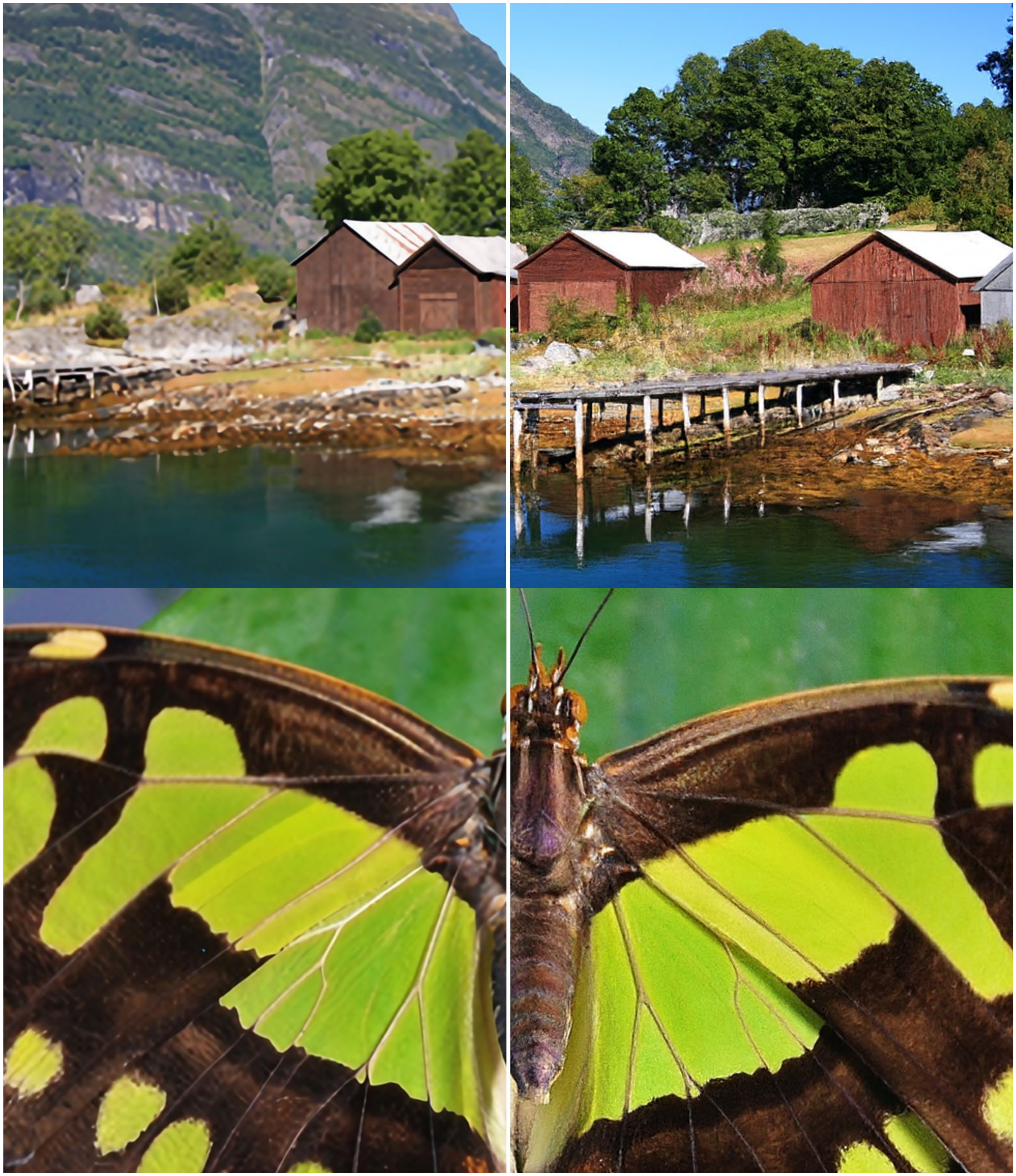

Figure 6. Additional visual result of super-resolution images. The left parts of the images are generated by a Regression-based method, the right parts of the images are generated by the proposed LAR-SR method.