Supplementary Material for "Content-Aware Local GAN for Photo-Realistic Super-Resolution"

JoonKyu Park[1] Sanghyun Son[1] Kyoung Mu Lee[1,2]
[1]Dept. of ECE&ASRI, [2]IPAI, Seoul National University, Korea
{jkpark0825, thstkdgus35, kyoungmu}@snu.ac.kr

(a) Bicubic (b) **CAL-GAN**

Figure S1: **Our CAL-GAN utilizes distribution-aware classifiers during the adversarial training and successfully super-resolves real-world low-resolution images.** Input images are crawled from Google.

S1. Introduction

Our CAL-GAN introduces a novel concept of *content-aware local adversarial learning* and reconstructs photo-realistic SR images from inputs as shown in Figure S1. In Section S2, we provide implementation details and efficient implementation of our discriminator architecture. In Section S3 and S4, we describe details about ablation studies in Section 4 of our main manuscript and provide additional ablation studies. In Section S5, we provide more visual comparisons to demonstrate the superiority of our CAL-GAN compared to the other perceptual SR models. We also note that all references are shared with our main manuscript.

S2. Discriminator Architecture

S2.1. Building blocks

Figure S2 shows the overall architecture of our CAL-GAN again. As explained in the main manuscript, our discriminator takes a high-resolution image $\mathbf{I}_{\text{HR}}$ or super-resolved image $\mathbf{I}_{\text{SR}}$ as its input and classifies each local feature whether it is drawn from real or fake distributions. Let us describe the discriminator architecture in detail. For simplicity, we denote both $\mathbf{I}_{\text{HR}}$ and $\mathbf{I}_{\text{SR}}$, and their corresponding features $\mathbf{F}_{\text{HR}}$ and $\mathbf{F}_{\text{SR}}$ as $\mathbf{I}_*$ and $\mathbf{F}_*$, respectively.

Feature extraction. We first extract $\mathbf{F}_*$ from the given input $\mathbf{I}_*$. The feature extractor consists of seven repeated Conv-BN-ReLU, where the second and sixth convolutions use stride of 2. Since the remaining convolutional layers use stride of 1, height and width of $\mathbf{F}_*$ are reduced by 1/4 compared to the original input $\mathbf{I}_*$.

Router architecture. We note that our routing module operates differently on HR feature $\mathbf{F}_{\text{HR}}$ and $\mathbf{F}_{\text{SR}}$. For an input HR feature $\mathbf{F}_{\text{HR}}$, the router first predicts the corresponding pixel-wise label. To this end, we employ a single 1×1 convolution without any activation function, as shown in Figure S2. The output of the convolutional layer has $N = 12$ channels, where N represents the number of the feature clusters or the classifiers. We apply Gumbel-Softmax across the channel dimension to construct N disjoint spatial binary masks $\mathbf{R}_i$, as described in (2) in our main manuscript. For $\mathbf{F}_{\text{SR}}$, we use the binary mask generated from its corresponding HR features $\mathbf{F}_{\text{HR}}$.

Orthogonal convolutions and classifiers. We employ N orthogonal convolutions to project N disjoint feature clusters to separated domains. Note that orthogonal convolutions do not change the dimensions of each input. Then, we use N independent classifiers $\{\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_N\}$ to discriminate whether pixels in each input feature $\mathbf{F}_i$ looks realistic or not. Each classifier consists of two 1×1 convolutions and a ReLU activation in between. We note that our router and classifiers have similar architectures, while the classifier output has a *single* channel only.

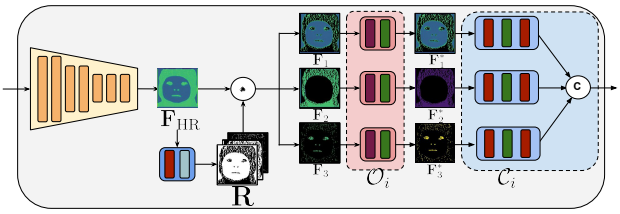


Figure S2: **Overall discriminator architecture of the proposed CAL-GAN framework.**

S2.2. Efficient implementation

Compared to the single-classifier configuration, our multi-classifier formulation requires additional computations for determining the real-fake labels of each local output. Specifically, the discriminator output $\mathbf{D}$ is a summation of N classifier outputs $\mathcal{C}_i\left(\mathbf{F}_i^*\right)$ as described in (8) of our main manuscript. To reduce the training overhead, we introduce the more efficient implementation of the multi-classifier system. Rather than performing dense spatial convolutions on $\mathbf{F}_i^*$, we first compose a dense vector $\hat{\mathbf{F}}_i^* \in \mathbb{R}^{c \times n_i}$ by collecting pixels from $\mathbf{F}_i^*$, which satisfy $\mathbf{R}_i\left[y, x\right] = 1$. We note that $n_i = \sum_{y,x} \mathbf{R}_i[y, x]$ is a number of valid pixels in the corresponding feature $\mathbf{F}_i^*$. As our classifier consists of 1×1 convolutions, it maps each c-dim input vector to a single scalar. Therefore, we can efficiently acquire pixel-wise labels $\hat{\mathbf{D}}_i \in \mathbb{R}^{1 \times n_i}$ by applying the classifier $\mathcal{C}_i$ to the rearranged features $\hat{\mathbf{F}}_i^*$. Finally, we construct the dense prediction $\mathbf{D}$ by gathering the predicted class-wise labels $\hat{\mathbf{D}}_i$ based on the original locations of pixels. We also note that such an implementation is for training time efficiency, and our discriminator is not used during inference.

S3. Details about our Ablation Study

S3.1. Details about model comparison

Table 1 of our main manuscript provides a quantitative comparison between the proposed CAL-GAN and other state-of-the-art approaches. Among them, all RRDB-based methods are based on Residual-in-Residual Dense Blocks. We note that ESRGAN, LDL, and our CAL-GAN share the same SR model architecture. On the other hand, SPSR [32] utilizes an additional gradient branch to estimate the translation of gradient maps. USRGAN [57] adopts the concept of iterative optimization by deep unfolding network (USRNet). Table S1 shows that CAL-GAN has the same computational cost as other RRDB-based models, while being more efficient than USRGAN and SPSR.

	ESRGAN	USRGAN	SPSR	LDL	**CAL-GAN**
MACs(G)	**294.24**	2431.79	869.24	**294.24**	**294.24**
Params(MB)	**16.7**	17.0	24.8	**16.7**	**16.7**

Table S1: **Computational efficiency comparison for the RRDB-based methods.** We note that ESRGAN, LDL, and our CAL-GAN share the same SR model for the generator. MACs are calculated for a 128×128 input image.

S3.2. Details about segmentation-based routing

Table 6 in our main manuscript compares the proposed learning-based routing strategy with semantic segmentation. For comparison, we subdivide the original 81 (80 objects + background) classes into 12 subsets. Here, we describe the details of the subdivision.

- 0: {background}
- 1: {person}
- 2: {bicycle, car, motorcycle, airplane, bus, train, truck, boat}
- 3: {traffic light, fire hydrant, stop sign, parking meter}
- 4: {bench, chair, couch, potted plant}
- 5: {bird, cat, dog, horse, sheep, cow, elephant, bear, zebra, giraffe, teddy bear}
- 6: {backpack, umbrella, handbag, tie, suitcase, clock}
- 7: {frisbee, sports ball, baseball bat, baseball glove, tennis racket, cell phone}
- 8: {skis, snowboard, skateboard, surfboard, kite}
- 9: {bottle, book, wine glass, cup, fork, knife, spoon, bowl, vase, scissors, hair drier, toothbrush}
- 10: {banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake}
- 11: {bed, dining table, toilet, tv, laptop, mouse, remote, keyboard, microwave, oven, toaster, sink, refrigerator}

S4. Additional Ablation Study

In this Section, we introduce in-depth ablation studies of our CAL-GAN that are not covered in our main manuscript.

Weight of the generator loss. A standard adversarial training framework alternately updates the generator and discriminator by the corresponding loss terms. In the proposed CAL-GAN, the SR model is updated by optimizing three independent loss terms, as described in (1) of our main manuscript. Here, we investigate the effect of the generator loss $\mathcal{L}_{\text{gen}}$, by varying its weight λ_{g}, as the term can dramatically affect the stability of the overall training process. Since our CAL-GAN achieves the best LPIPS with $\lambda_{\text{g}} = 0.005$, we use the hyperparameter value by default throughout our main manuscript.

Capacity of the discriminator. In our proposed MoC, specialized discriminators are utilized on each local content, and the capacity of the discriminator can impact the overall performance of CAL-GAN. We conduct an ablation study to analyze the effect of discriminator capacity by utilizing a U-Net-based discriminator or using our discriminator architecture with double the number of channels (Original$\times$2), as presented in Table S3. The results indicate a slight improvement in performance when using a larger discriminator. However, the U-Net-based discriminator did not perform well despite its larger capacity. This may be due to the U-Net architecture having a much wider receptive field than ours, making it challenging to assign dense pixel-wise categories at the output side.

λ_{g}	**0.005**	0.010	0.020	0.050	0.100
LPIPS	**0.091**	0.093	0.093	0.097	0.105

Table S2: **Quantitative comparisons between our CAL-GAN with different generator loss weights λ_{g}.**

Discriminator architecture	Params (M)	LPIPS$_{\downarrow}$	FID$_{\downarrow}$	BRISQUE$_{\downarrow}$	DISTS$_{\downarrow}$
Original	2.68	**0.091**	11.772	13.576	0.049
Original$\times$2	5.34	**0.091**	**11.493**	**12.777**	**0.048**
U-Net-based	4.38	0.094	13.480	14.349	0.052

Table S3: **Effects of discriminator network architecture on DIV2K (val).**

Comparison with LDL with various backbone networks. Table 1 in the main paper includes a comparison with the RRDB baseline. In addition to that, we provide a comparison using EDSR and SwinIR as baseline networks in Table S4, while utilizing the second-best method, LDL. The results demonstrate that CAL-GAN significantly enhances the perceptual super-resolution performance across various metrics.

Baseline	Method	LPIPS$_{\downarrow}$	FID$_{\downarrow}$	BRISQUE$_{\downarrow}$	DISTS$_{\downarrow}$
EDSR	+ LDL	0.081	16.352	11.850	**0.053**
	+CAL-GAN	**0.074**	**15.863**	**11.555**	0.078
SwinIR	+ LDL	0.094	**12.075**	11.852	0.051
	+CAL-GAN	**0.087**	12.097	**11.406**	**0.048**

Table S4: **Further comparison with LDL on DIV2K (val).**

Distribution of routing mask. In our main paper, we utilized 12 classes to route local content to different classes. In this work, we introduce a novel balancing loss $\mathcal{L}_{\text{b}}$ to ensure an even distribution of classes. Figure S3 displays a histogram of the 12 classifiers, demonstrating that our router evenly routes local content into different classes.

CAL-GAN for $\times 2$ SR. Our main manuscript provides analysis on $\times 4$ CAL-GAN only. Here, we also compare $\times 2$ CAL-GAN with the other representative SR models.

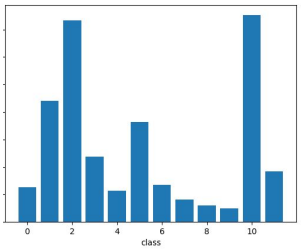


Figure S3: **Distribution of routing mask.**

As shown in Table S5, our CAL-GAN achieves comparable performance with the other state-of-the-art methods on $\times 2$ SR task as well.

Metric	ESRGAN	LDL	**CAL-GAN**
LPIPS$_\downarrow$	0.0261	**0.0260**	**0.0260**
DISTS$_\downarrow$	**0.0142**	0.0143	0.143
FID$_\downarrow$	4.856	4.775	**4.740**

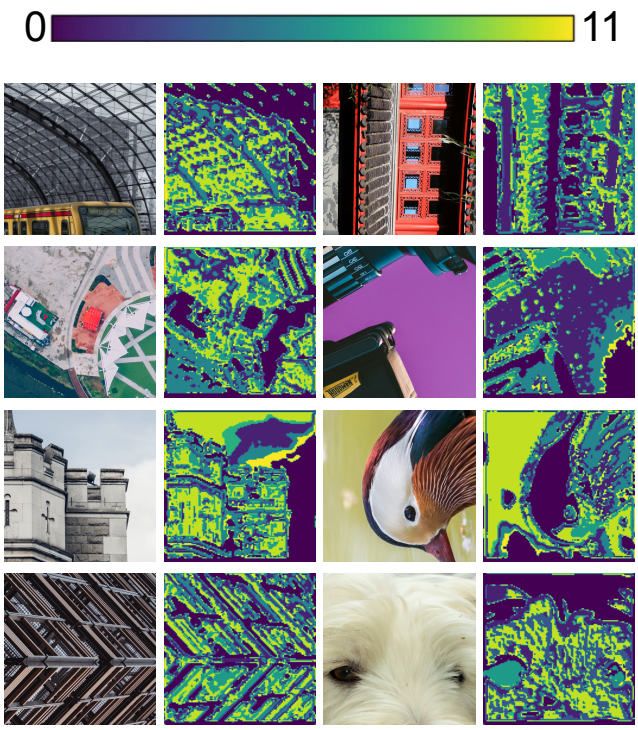
Table S5: **Comparison with photo-realistic SR methods on $\times 2$ SR (DIV2K (val)).** Metrics for ESRGAN and LDL are referred from [29].

Non-reference metrics on real-world images. Besides providing visual results on real-world images in Figure 7 of our main manuscript, we provide a quantitative comparison in Table S6. Since the high-resolution counterpart image is not prepared for the real images, we show the non-reference metrics in Table S6. The table demonstrates that our CAL-GAN achieves BRISQUE comparable to BSRGAN and SoTA NIQE, outperforming other methods.

Metrics	RealESRGAN [48]	BSRGAN [58]	**CAL-GAN**
BRISQUE$_\downarrow$	5.77	**5.60**	**5.60**
NIQE$_\downarrow$	18.01	20.16	**16.18**

Table S6: **Non-reference metrics comparison on real-world RealSRSet [58] dataset.**

Routing mask visualization. In the main manuscript, we present Figures 3 and 5, which show the routing mask $\mathbf{R}$ using a limited number of colors for simplicity. However, we provide a more detailed representation of the mask in Figure S4 using a full range of 12 colors and a corresponding color bar. Notably, unlike segmentation-based methods that cannot distinguish between different classes of the same object, our router can differentiate local content within the same object. For instance, in the image in row 3, column 2, our router accurately classifies a single object (*e.g.*, a bird) into multiple classifiers.



(b) Input (c) Routing mask (d) Input (e) Routing mask

Figure S4: **Visualization of the routing mask.**

S5. Additional Visual Comparison

We provide more qualitative results from our CAL-GAN in Figure S5, S6, S7, S8, and S9, evaluated on synthetic LR images. Also, Figure S10 and S11 illustrate the results of our CAL-GAN applied on real-world LR images. The results clearly demonstrate that the proposed CAL-GAN produces perceptually better outputs on various types of inputs compared to the existing methods.

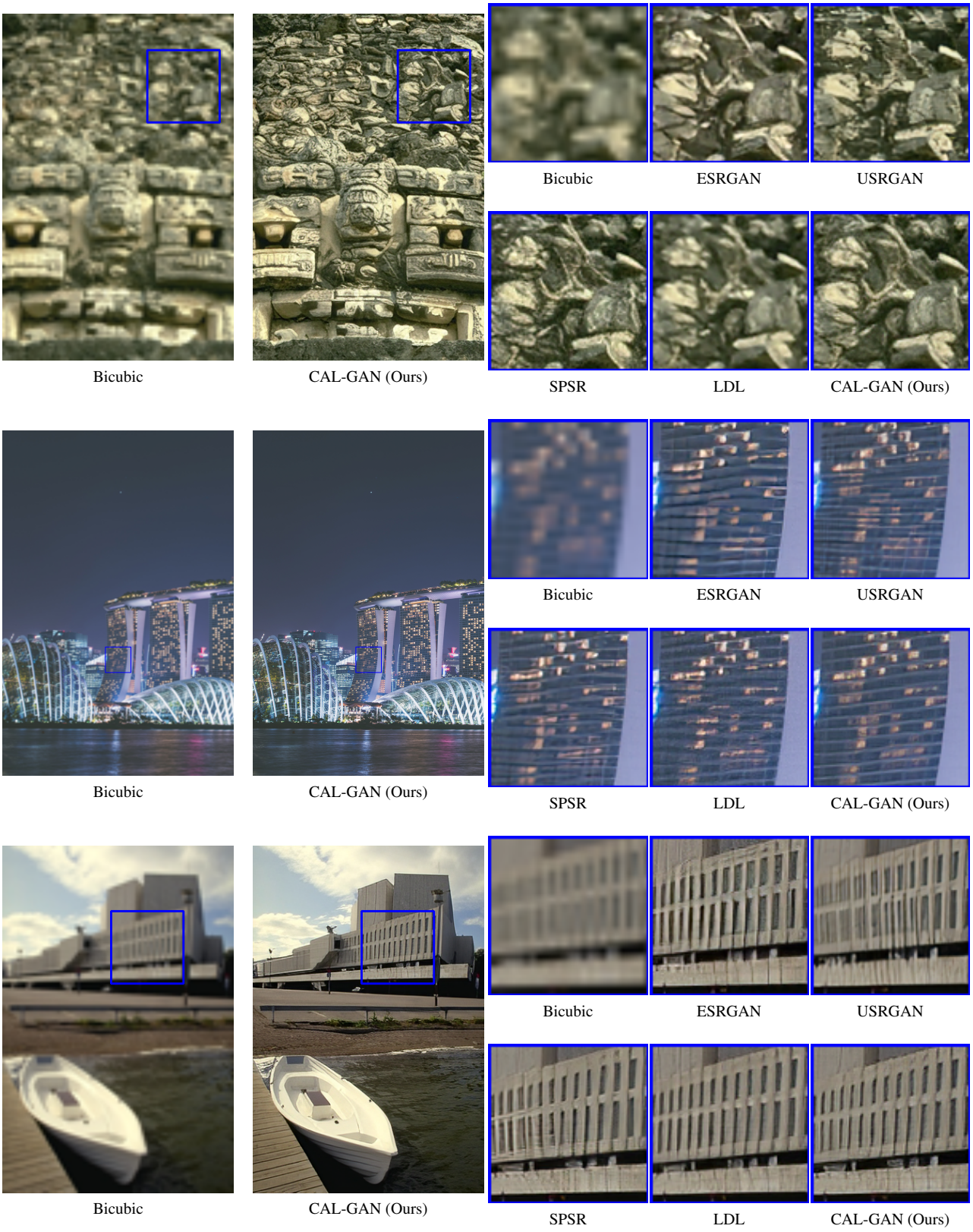


Figure S5: **Qualitative comparison with state-of-the-art methods on synthetic datasets (1).** Please zoom in for better details.

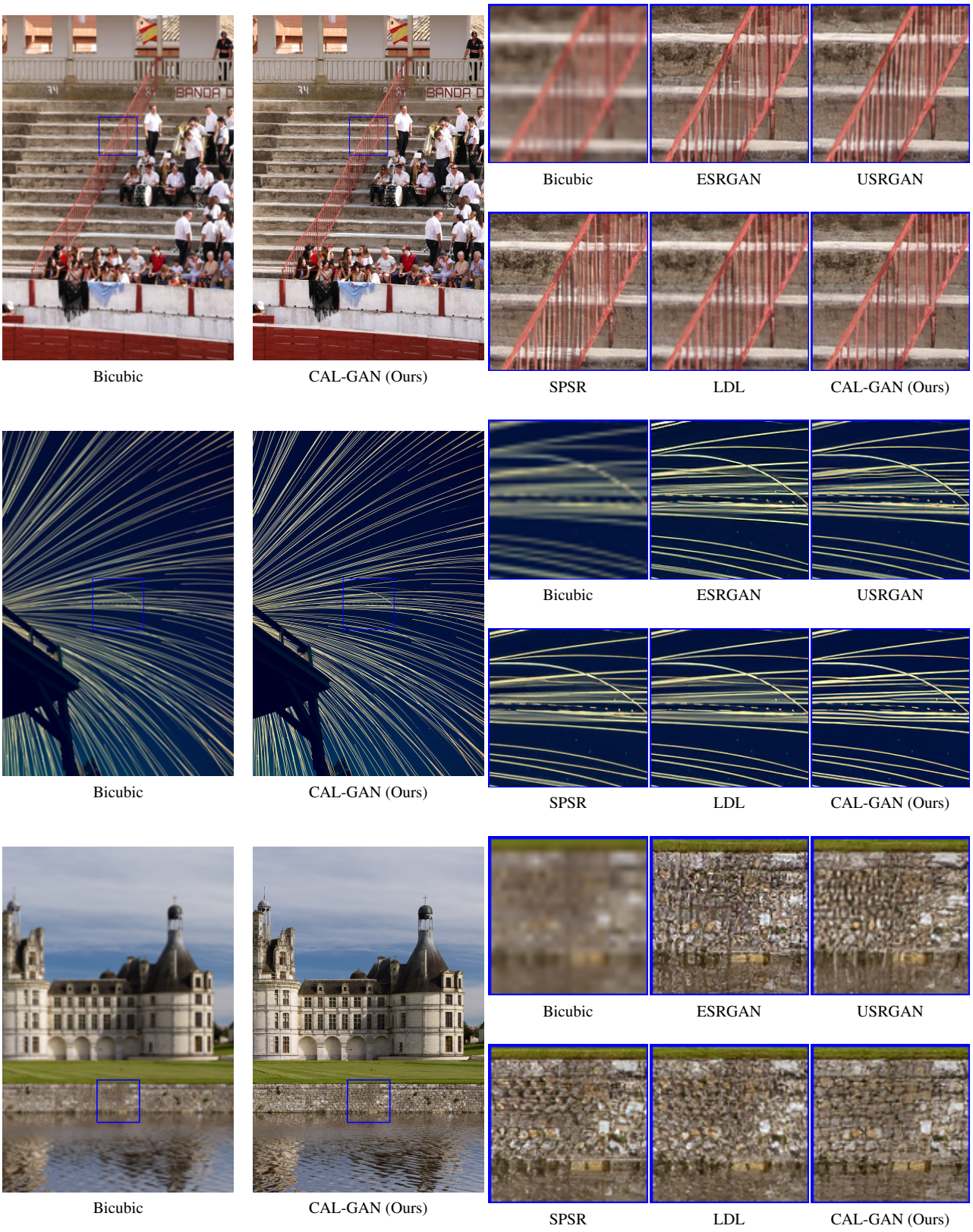


Figure S6: **Qualitative comparison with state-of-the-art methods on synthetic datasets (2).** Please zoom in for better details.

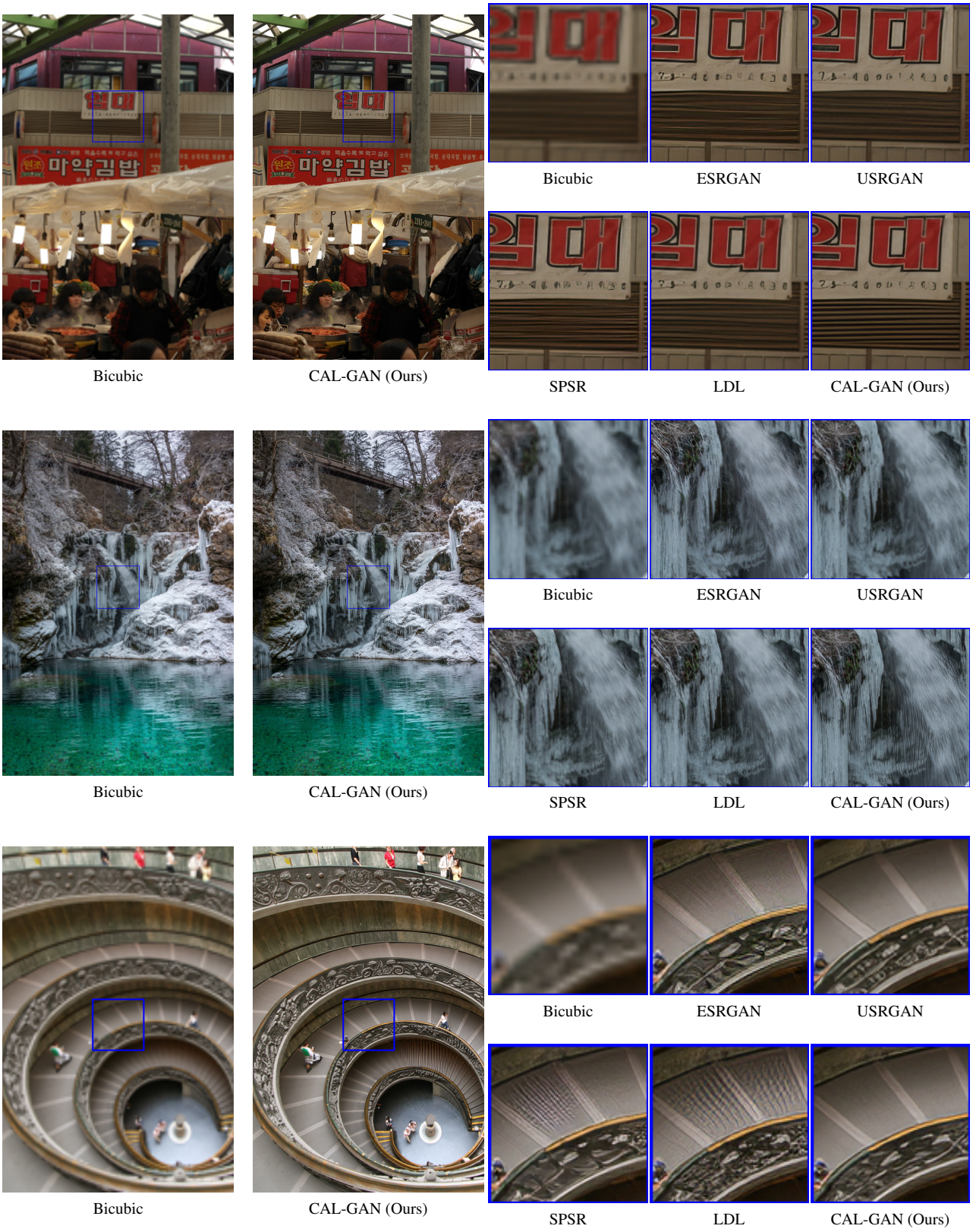


Figure S7: **Qualitative comparison with state-of-the-art methods on synthetic datasets (3).** Please zoom in for better details.

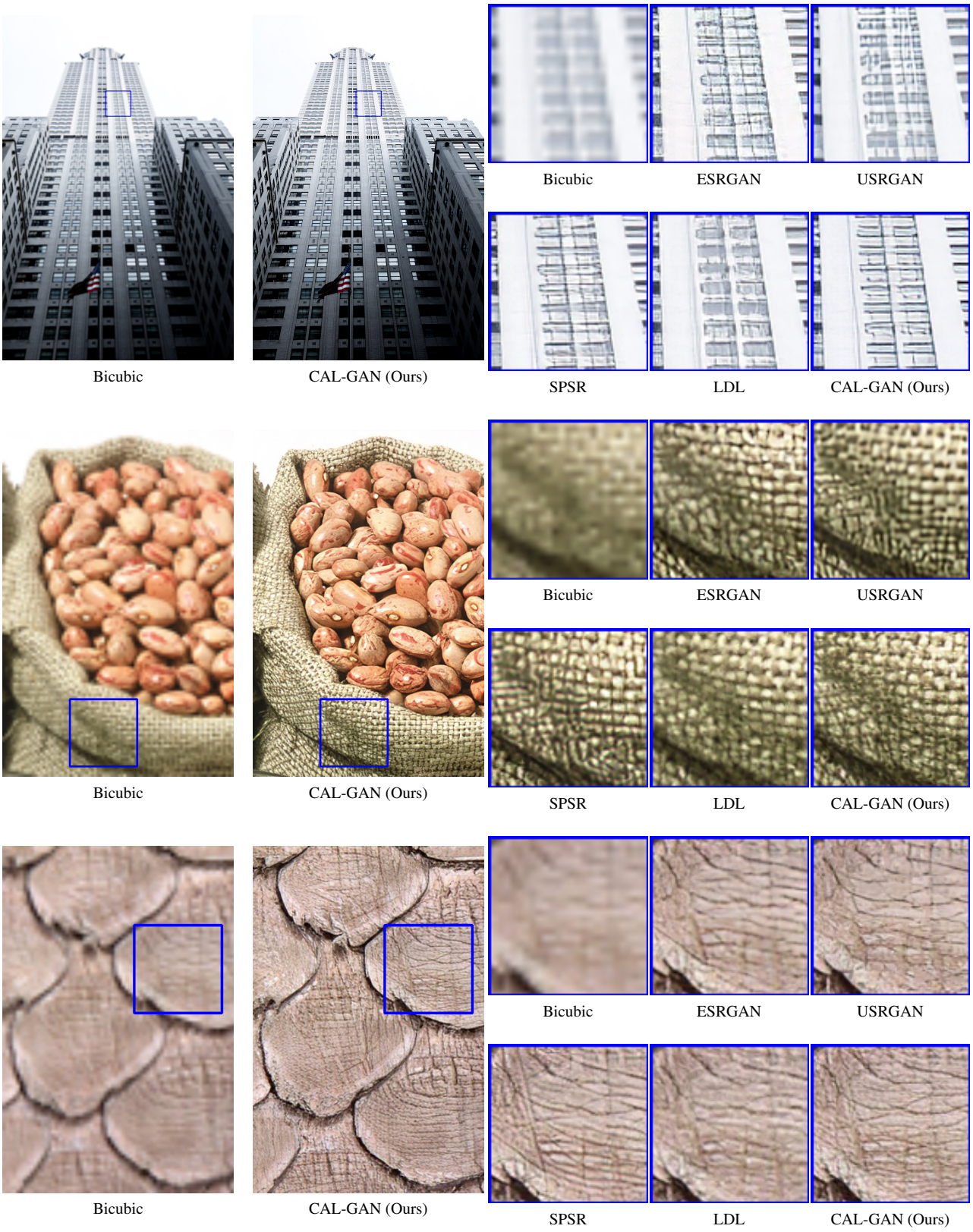


Figure S8: **Qualitative comparison with state-of-the-art methods on synthetic datasets (4).** Please zoom in for better results.

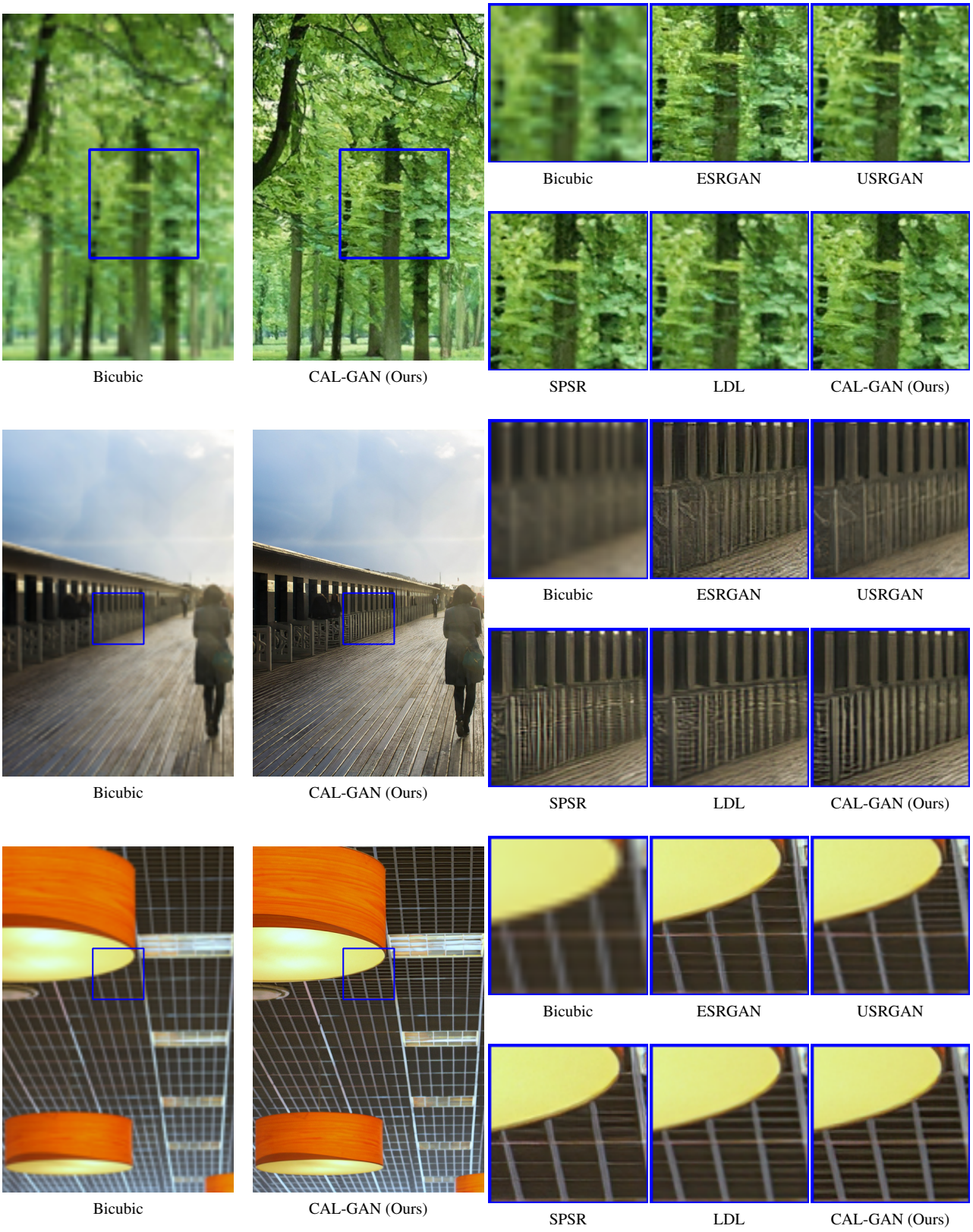


Figure S9: **Qualitative comparison with state-of-the-art methods on synthetic datasets (5).** Please zoom in for better results.

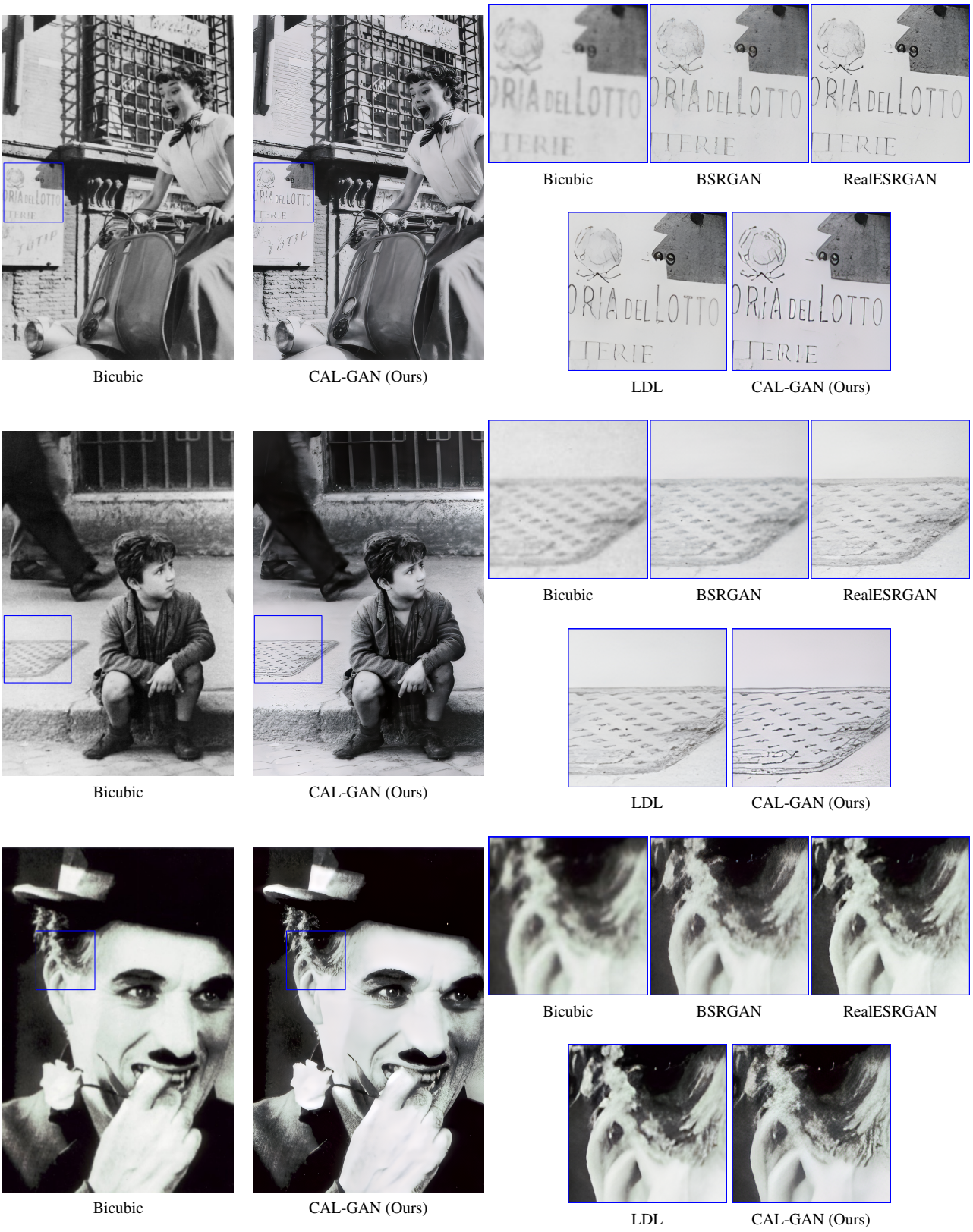


Figure S10: **Qualitative comparison with state-of-the-art methods on real-world images (1).** The images are crawled from Google.

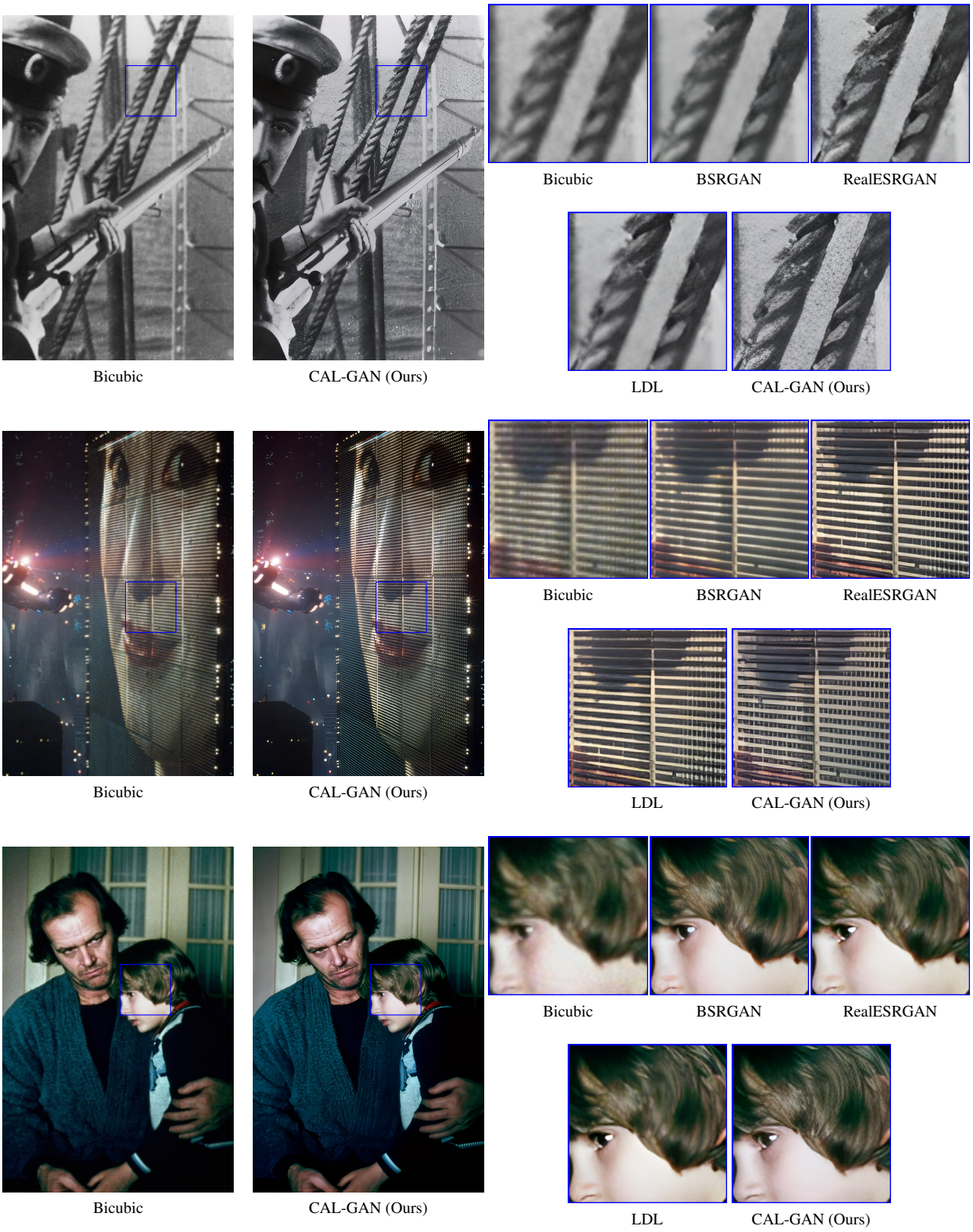


Figure S11: **Qualitative comparison with state-of-the-art methods on real-world images (2).** The images are crawled from Google.