MI-GAN: A Simple Baseline for Image Inpainting on Mobile Devices

Andranik Sargsyan[1], Shant Navasardyan[1], Xingqian Xu[1,2], Humphrey Shi[1,2]
[1]Picsart AI Research (PAIR), [2]SHI Labs @ Georgia Tech, Oregon & UIUC

Device Name	512-resolution	
	MI-GAN speed (ms, mean/std)	Co-Mod-GAN speed (ms, mean/std)
iPhone7	1855.60 / 29.69	6254.60 / 39.43
iPhoneX	1131.00 / 11.14	3943.33 / 55.89
iPad mini (5th gen)	1017.67 / 28.87	3767.71 / 95.72
iPhone14-pro-max	539.10 / 17.89	1880.00 / 55.92
Galaxy Tab S7+	1210.50 / 30.84	- / -
Samsung Galaxy S8	2527.47 / 40.70	- / -
vivo Y12	5385.67 / 313.95	- / -

Table S.1: The actual speed of our 512 resolution model vs Co-Mod-GAN [4] 512 resolution model deployed on various mobile devices. The actual speed is measured in miliseconds, the means and the standard deviations of the speed are presented for each mobile device.

1. Network Architecture Details

In this section we present the architectural details of our MI-GAN generator and discriminator.

Table S.4 presents the detailed layer architecture of 256 and 512 resolution MI-GAN models. As can be seen from the table, our 512x512 model has one more downsampling and upsampling layers compared to our 256 model. These additional layers serve for two main purposes: first, they help to decrease the computational cost and memory usage with 512x512 inputs, and second, they increase the receptive field of the network to better handle larger holes.

MI-GAN discriminator mainly consists of residual blocks containing depthwise-separable convolution layers and bilinear downsampling operations. Similar to the MI-GAN generator, we apply re-parametrization trick to the discriminator in order to increase the discriminative ability of the network. All discriminator layers, except the skip connection layers and the output layer, use leaky ReLU activation function with $\alpha = 0.2$. Activation functions for residual connection layers and the output layer are linear. Similar to [2] we use a minibatch standard deviation layer near the end of convolution layers of the discriminator. Please see Table S.5 for a more detailed representation of the discriminator architecture.

2. 512 Model Speed on Mobile Devices

As can be noticed from Table S.1, on actual mobile devices, with 512x512 inputs, our 512 resolution model runs on average 3.5 times faster than the corresponding Co-Mod-GAN [4] model. On some test devices Co-Mod-GAN failed to run due to a memory error. The fact that our 512 resolution model can run on various mobile devices, with a reasonable speed and almost state-of-the-art quality, make it feasible to be used in real-world applications.

3. More Quantitative Results

In addition to quantitative comparisons done in the paper, this section presents extra metrics which show the robustness of our approach to different mask sizes and metric selections. Table 2 presents FID metrics calculated on different mask area ranges for Places 2 [5] and FFHQ [1] datasets. The results show that our method is comparable or sometimes better than other SOTAs for large as well as for small mask ratios. Table 3 presents the PSNR, SSIM [3], P-IDS and U-IDS [4] metrics calculated on Places 2 and FFHQ datasets. The table shows that MI-GAN is comparable or better according to those metrics as well.

4. More Qualitative Results

In this section we present additional set of qualitative results of our model in comparison with the results of state-of-the-art approaches. These results additionally support the claims made in Section 4.3 of the paper. Figure S.1 compares MI-GAN results with other approaches results on 256x256 resolution Places 2 [5] images. Figures S.2 and S.3 show more results on 512x512 resolution Places 2 images. Figures S.4 and S.5 provide more results on FFHQ dataset [1] samples. Figures S.6 and S.7 present additional results with object masks. Figure S.8 includes more samples from our user study.

5. Failure Cases

As mentioned in the Section 5 of the paper, our approach meets difficulties when it comes to reconstructing complex 3D scenes. In Figure S.9 we present sample results of such

Method	**Places 2** (512x512) Mask area range (0, .2)	(.2, .4)	(.4, .6)	(.6, .8)	(.8, 1)	**FFHQ** (256x256) Mask area range (0, .2)	(.2, .4)	(.4, .6)	(.6, .8)	(.8, 1)
LDM	0.29	1.37	3.89	9.72	21.92	-	-	-	-	-
SH-GAN	0.32	1.60	4.22	8.35	13.64	0.68	1.85	3.03	4.32	5.63
Co-Mod-GAN	0.34	1.79	4.80	9.74	15.45	0.69	1.91	3.27	4.76	6.22
MAT	0.40	1.75	4.67	10.36	19.10	0.58	1.92	3.95	6.99	10.35
ZITS	0.32	1.54	4.74	13.88	39.78	-	-	-	-	-
LaMa	0.29	1.84	6.42	18.35	46.23	0.58	1.92	3.95	6.99	10.35
HiFill	0.92	8.16	34.02	85.19	128.20	-	-	-	-	-
MI-GAN (ours)	0.36	2.03	5.74	12.19	20.51	0.71	1.99	3.38	5.04	6.55

Table 2: FID comparison with varying mask area ranges on Places2-512 and FFHQ-256. The table shows that our approach is comparable to SOTA heavy approaches on both datasets and better than HiFill on all mask area ranges.

Method	**Places 2** (512x512) **PSNR** ↑	**SSIM** ↑	**P-IDS** ↑	**U-IDS** ↑	**FFHQ** (256x256) **PSNR** ↑	**SSIM** ↑	**P-IDS** ↑	**U-IDS** ↑
LDM	16.106	0.605	0.083	0.237	-	-	-	-
SH-GAN	16.014	0.597	0.149	0.286	16.357	0.591	0.154	0.272
Co-Mod-GAN	15.985	0.595	0.120	0.266	16.247	0.591	0.137	0.262
MAT	15.934	0.597	0.104	0.249	17.080	0.627	0.041	0.170
ZITS	18.131	0.657	0.033	0.171	-	-	-	-
LaMa	17.950	0.655	0.020	0.143	17.585	0.628	0.002	0.015
HiFill	15.599	0.544	0.002	0.044	-	-	-	-
MI-GAN (ours)	16.118	0.597	0.091	0.234	16.315	0.590	0.119	0.239

Table 3: Additional metrics on Places2-512 and FFHQ-256. Results show that, according to PSNR, SSIM, P-IDS and U-IDS metrics, our approach is still comparable or better than bigger and slower SOTA methods. Our approach is better than HiFill on all qualitative metrics in comparison.

failure cases. For example, in the first column of the figure, our model fails to reconstruct the billiard table and the tennis table, and in the second column our model fails to realistically complete the telephone booth.

References

[1] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405, 2019. 1

[2] Yuri Viazovetskyi, Vladimir Ivashkin, and Evgeny Kashin. Stylegan2 distillation for feed-forward image manipulation. In *European conference on computer vision*, pages 170–186. Springer, 2020. 1

[3] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402 Vol.2, 2003. 1

[4] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2021. 1

[5] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 1

Table S.4: MI-GAN generator architecture details. *dw-sep* denotes the depthwise-separable layer described in Section 3.1 of the paper. Notations *ds* and *us* mean bilinear downsample and bilinear upsample respectively. Notation $noise$ indicates that noise addition is being applied inside the depthwise separable convolution block. $torgb$ layers are conv1×1 layers with linear activation functions. Those layers form a parallel branch, the Painting branch, which converts deep features into RGB images, and aggregates the results through bilinear upsampling and element-wise addition, as presented in the paper.

	Input resolution	256 resolution model	512 resolution model
Encoder	512×512		conv1×1 ($4 \rightarrow 64$) dw-sep 3×3 ($64 \rightarrow 64$) dw-sep 3×3 ($64 \rightarrow 128$, ds)
	256×256	conv1×1 ($4 \rightarrow 128$) dw-sep 3×3 ($128 \rightarrow 128$) dw-sep 3×3 ($128 \rightarrow 256$, ds)	dw-sep 3×3 ($128 \rightarrow 128$) dw-sep 3×3 ($128 \rightarrow 256$, ds)
	128×128	dw-sep 3×3 ($256 \rightarrow 256$) dw-sep 3×3 ($256 \rightarrow 512$, ds)	dw-sep 3×3 ($256 \rightarrow 256$) dw-sep 3×3 ($256 \rightarrow 512$, ds)
	64×64	dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)	dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)
	32×32	dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)	dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)
	16×16	dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)	dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)
	8×8	dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)	dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)
	4×4	dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$)	dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$)
Decoder	4×4	dw-sep 3×3 ($512 \rightarrow 512$, noise) dw-sep 3×3 ($512 \rightarrow 512$, noise) torgb ($512 \rightarrow 3$)	dw-sep 3×3 ($512 \rightarrow 512$, noise) dw-sep 3×3 ($512 \rightarrow 512$, noise) torgb ($512 \rightarrow 3$)
	8×8	dw-sep 3×3 ($512 \rightarrow 512$, us, noise) dw-sep 3×3 ($512 \rightarrow 512$, noise) torgb ($512 \rightarrow 3$)	dw-sep 3×3 ($512 \rightarrow 512$, us, noise) dw-sep 3×3 ($512 \rightarrow 512$, noise) torgb ($512 \rightarrow 3$)
	16×16	dw-sep 3×3 ($512 \rightarrow 512$, us, noise) dw-sep 3×3 ($512 \rightarrow 512$, noise) torgb ($512 \rightarrow 3$)	dw-sep 3×3 ($512 \rightarrow 512$, us, noise) dw-sep 3×3 ($512 \rightarrow 512$, noise) torgb ($512 \rightarrow 3$)
	32×32	dw-sep 3×3 ($512 \rightarrow 512$, us, noise) dw-sep 3×3 ($512 \rightarrow 512$, noise) torgb ($512 \rightarrow 3$)	dw-sep 3×3 ($512 \rightarrow 512$, us, noise) dw-sep 3×3 ($512 \rightarrow 512$, noise) torgb ($512 \rightarrow 3$)
	64×64	dw-sep 3×3 ($512 \rightarrow 512$, us, noise) dw-sep 3×3 ($512 \rightarrow 512$, noise) torgb ($512 \rightarrow 3$)	dw-sep 3×3 ($512 \rightarrow 512$, us, noise) dw-sep 3×3 ($512 \rightarrow 512$, noise) torgb ($512 \rightarrow 3$)
	128×128	dw-sep 3×3 ($512 \rightarrow 256$, us, noise) dw-sep 3×3 ($256 \rightarrow 256$, noise) torgb ($256 \rightarrow 3$)	dw-sep 3×3 ($512 \rightarrow 256$, us, noise) dw-sep 3×3 ($256 \rightarrow 256$, noise) torgb ($256 \rightarrow 3$)
	256×256	dw-sep 3×3 ($256 \rightarrow 128$, us, noise) dw-sep 3×3 ($128 \rightarrow 128$, noise) torgb ($128 \rightarrow 3$)	dw-sep 3×3 ($256 \rightarrow 128$, us, noise) dw-sep 3×3 ($128 \rightarrow 128$, noise) torgb ($128 \rightarrow 3$)
	512×512		dw-sep 3×3 ($128 \rightarrow 64$, us, noise) dw-sep 3×3 ($64 \rightarrow 64$, noise) torgb ($64 \rightarrow 3$)

Table S.5: MI-GAN discriminator architecture details. *dw-sep* denotes the depthwise-separable convolution layer described in Section 3.1 of the paper. Notation *ds* indicates the bilinear downsampling operation. *skip* is a 1×1 convolutional layer with a linear activation, through which a direct residual connection is made between each resolution block input and its output. We combine residual features and block output features through element-wise addition. *mbstd* denotes the minibatch standard deviation layer.

	Input resolution	256 resolution model	512 resolution model
Discriminator	512×512		conv1×1 ($4 \rightarrow 64$) skip ($64 \rightarrow 128$, ds) dw-sep 3×3 ($64 \rightarrow 64$) dw-sep 3×3 ($64 \rightarrow 128$, ds)
	256×256	conv1×1 ($4 \rightarrow 128$) skip ($128 \rightarrow 256$, ds) dw-sep 3×3 ($128 \rightarrow 128$) dw-sep 3×3 ($128 \rightarrow 256$, ds)	skip ($128 \rightarrow 256$, ds) dw-sep 3×3 ($128 \rightarrow 128$) dw-sep 3×3 ($128 \rightarrow 256$, ds)
	128×128	skip ($256 \rightarrow 512$, ds) dw-sep 3×3 ($256 \rightarrow 256$) dw-sep 3×3 ($256 \rightarrow 512$, ds)	skip ($256 \rightarrow 512$, ds) dw-sep 3×3 ($256 \rightarrow 256$) dw-sep 3×3 ($256 \rightarrow 512$, ds)
	64×64	skip ($512 \rightarrow 512$, ds) dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)	skip ($512 \rightarrow 512$, ds) dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)
	32×32	skip ($512 \rightarrow 512$, ds) dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)	skip ($512 \rightarrow 512$, ds) dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)
	16×16	skip ($512 \rightarrow 512$, ds) dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)	skip ($512 \rightarrow 512$, ds) dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)
	8×8	skip ($512 \rightarrow 512$, ds) dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)	skip ($512 \rightarrow 512$, ds) dw-sep 3×3 ($512 \rightarrow 512$) dw-sep 3×3 ($512 \rightarrow 512$, ds)
	4×4	mbstd ($512 \rightarrow 513$) dw-sep 3×3 ($513 \rightarrow 512$)	mbstd ($512 \rightarrow 513$) dw-sep 3×3 ($513 \rightarrow 512$)
		flatten ($512 \times 4 \times 4 \rightarrow 8192$) fc ($8192 \rightarrow 512$) fc ($512 \rightarrow 1$)	flatten ($512 \times 4 \times 4 \rightarrow 8192$) fc ($8192 \rightarrow 512$) fc ($512 \rightarrow 1$)

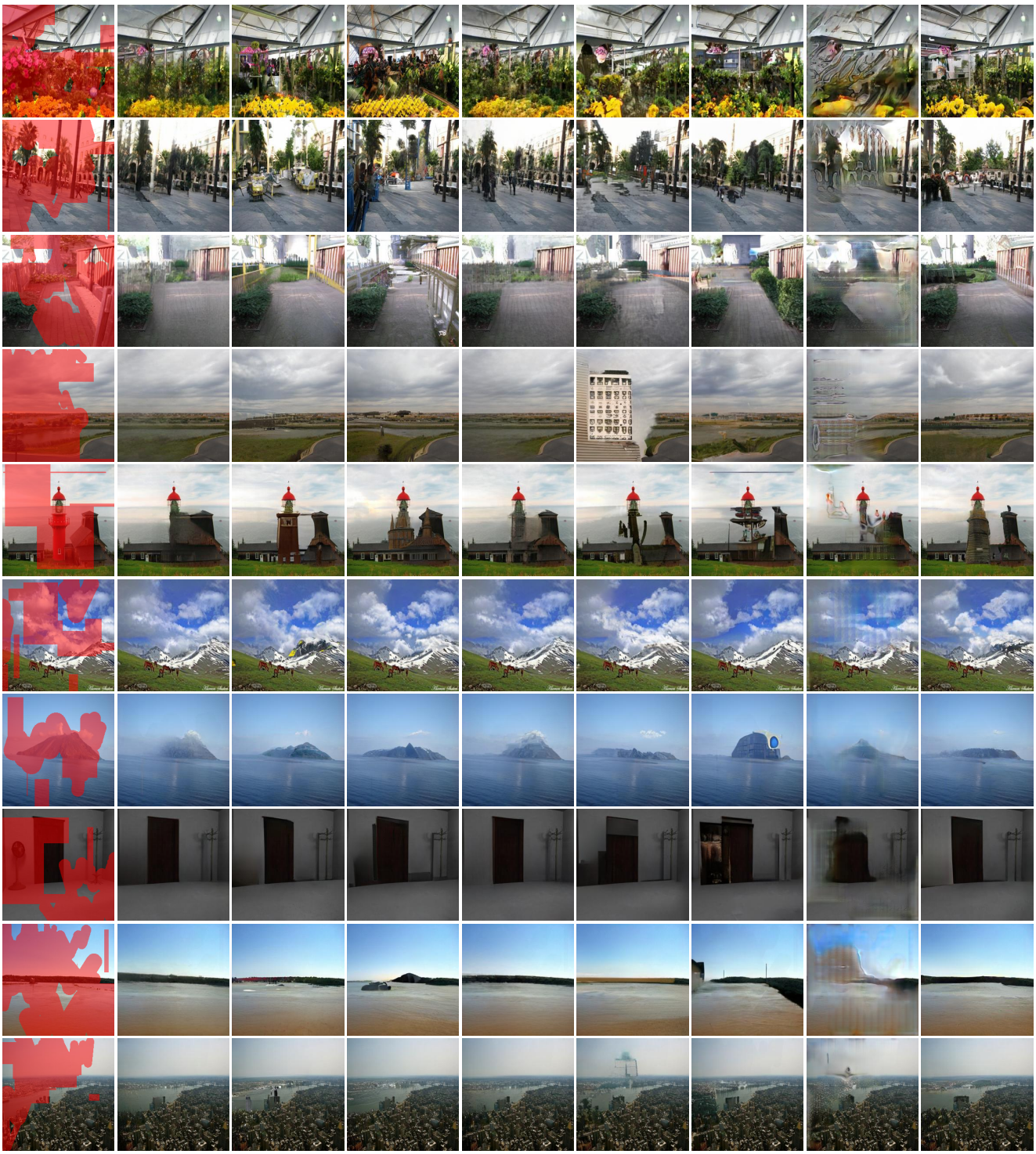


Figure S.1: Example results of our 256 resolution model and other state of the art approaches on 256x256 resolution Places2 samples using free-form masks.

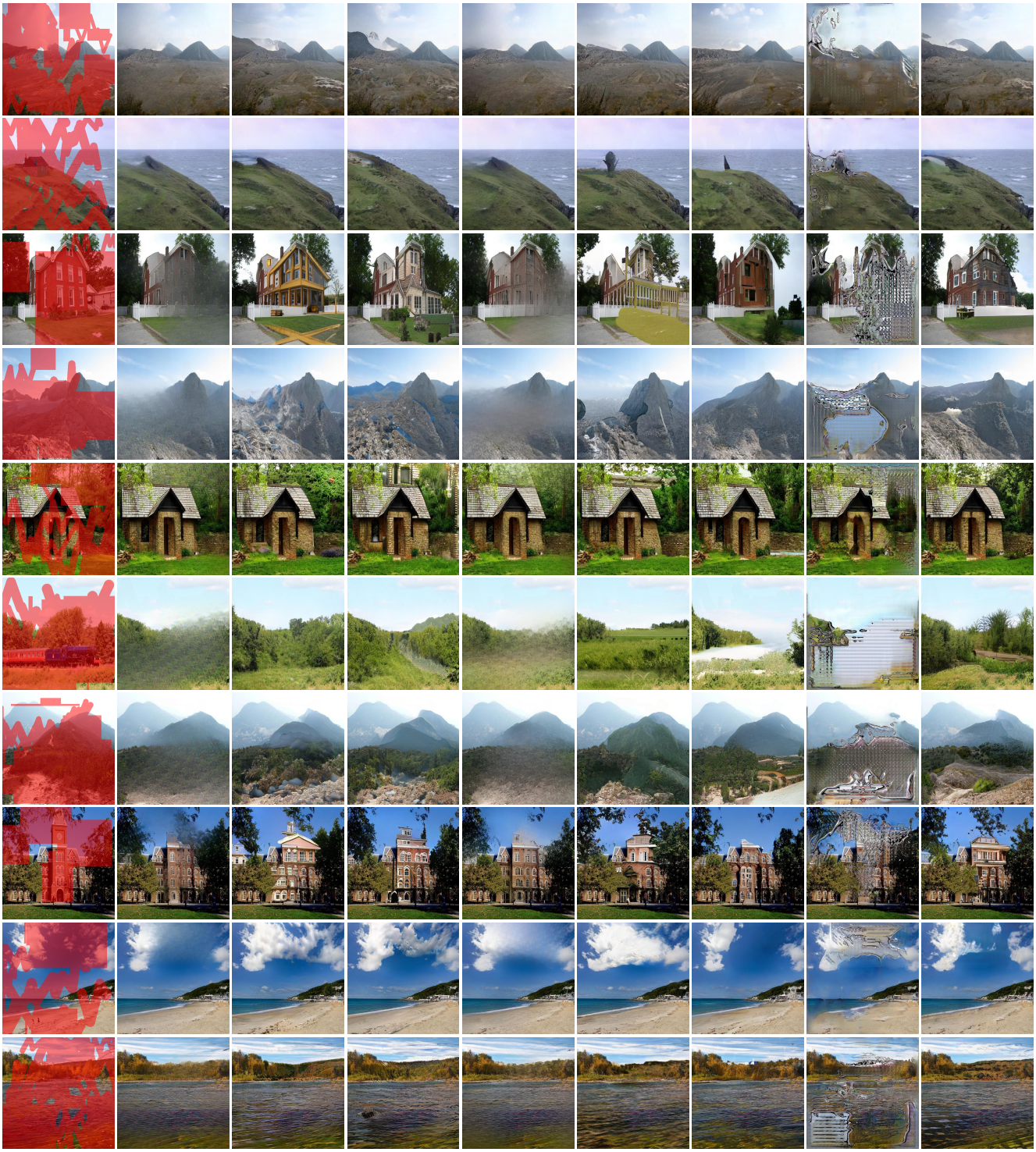


Figure S.2: Example results of our 512 resolution model and other state of the art approaches on 512x512 resolution Places2 samples using free-form masks. Please zoom for a better view.

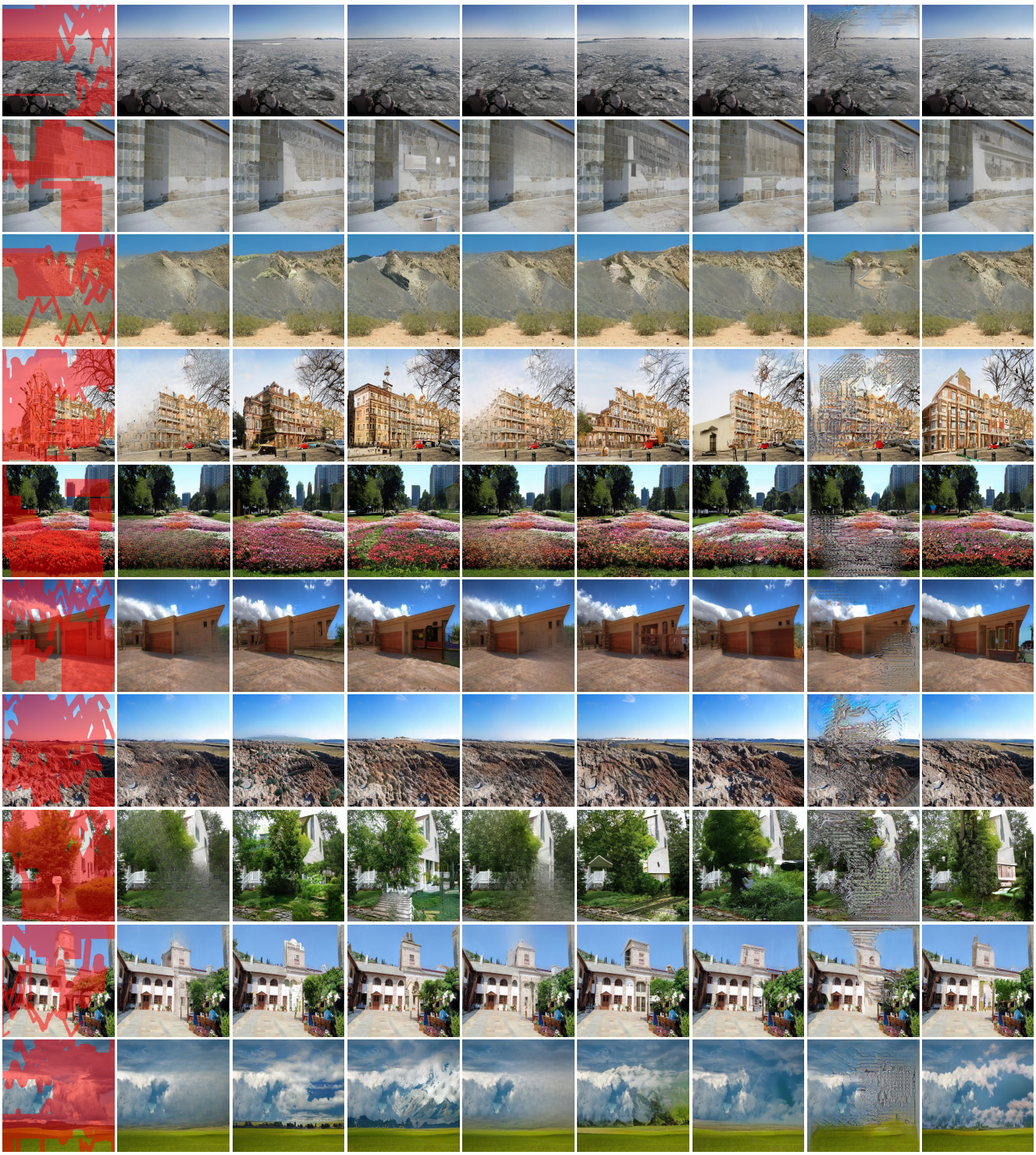


Figure S.3: Example results of our 512 resolution model and other state of the art approaches on 512x512 resolution Places2 samples using free-form masks. Please zoom for a better view.

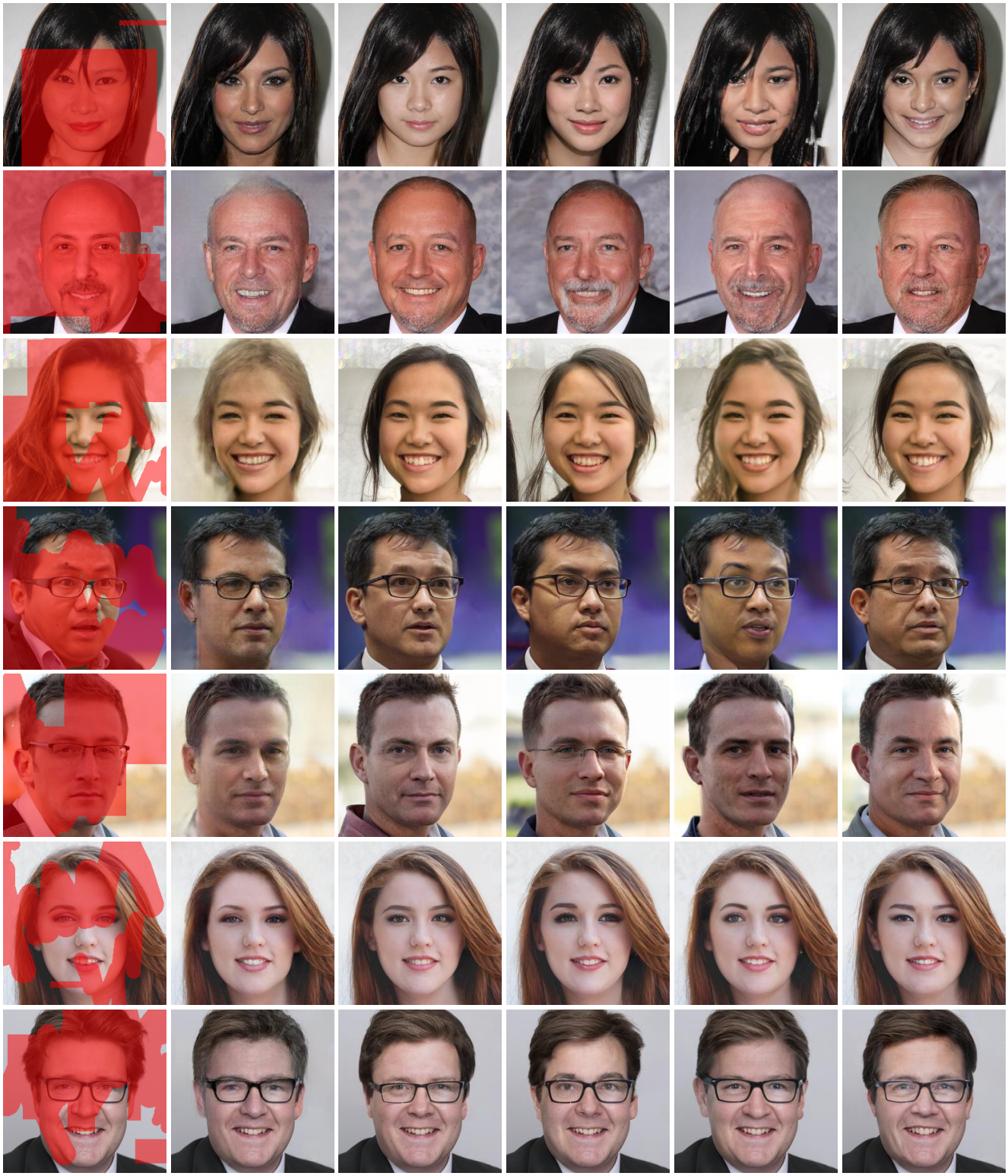
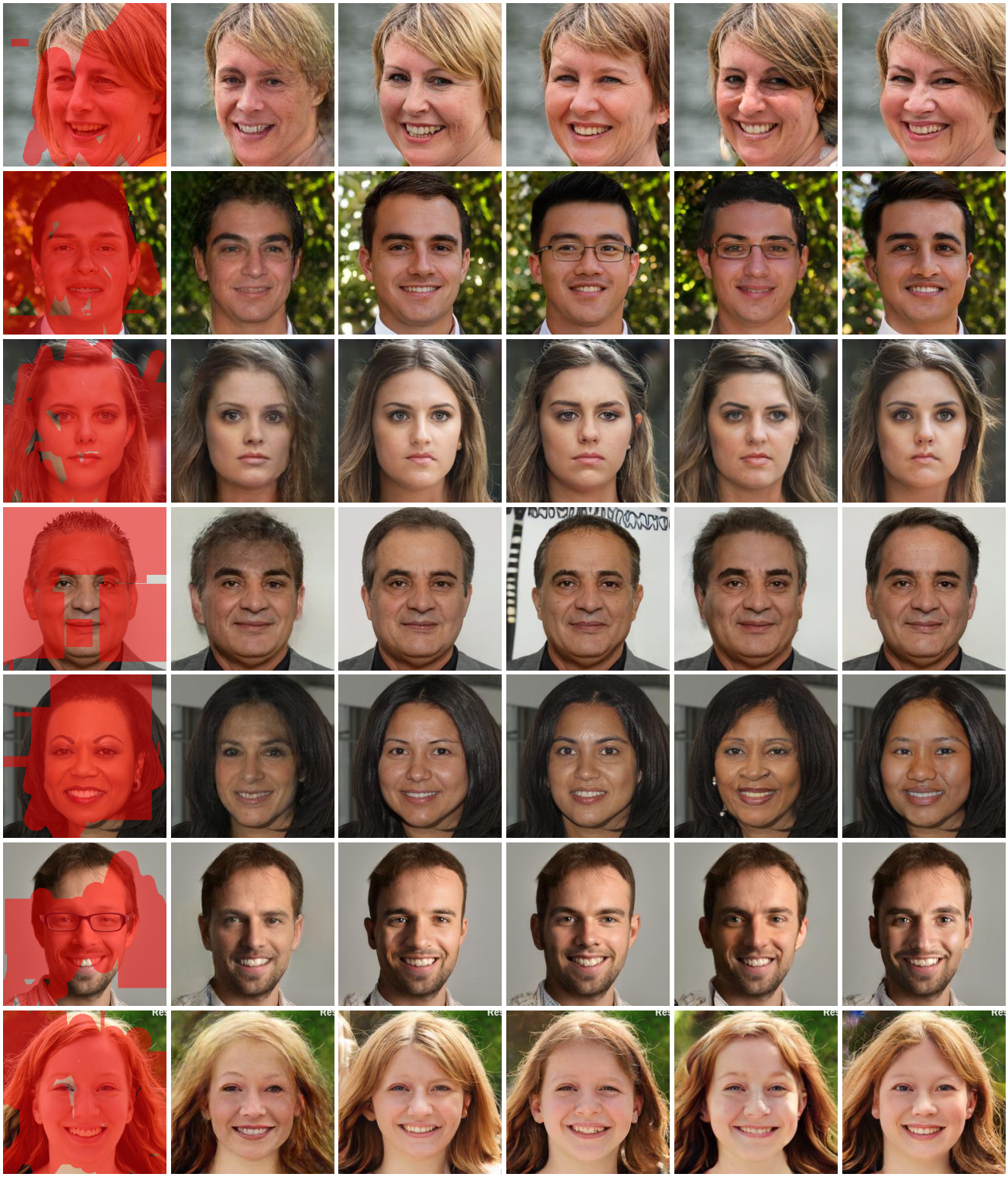


Figure S.4: Example results of our 256 resolution model and other state of the art approaches on 256x256 resolution FFHQ samples using free-form masks.



Masked image | LaMa | Co-Mod-GAN | SH-GAN | MAT | MI-GAN (ours)

Figure S.5: Example results of our 256 resolution model and other state of the art approaches on 256x256 resolution FFHQ samples using free-form masks.

Figure S.6: Additional results of our 512 resolution model and other approaches using object masks

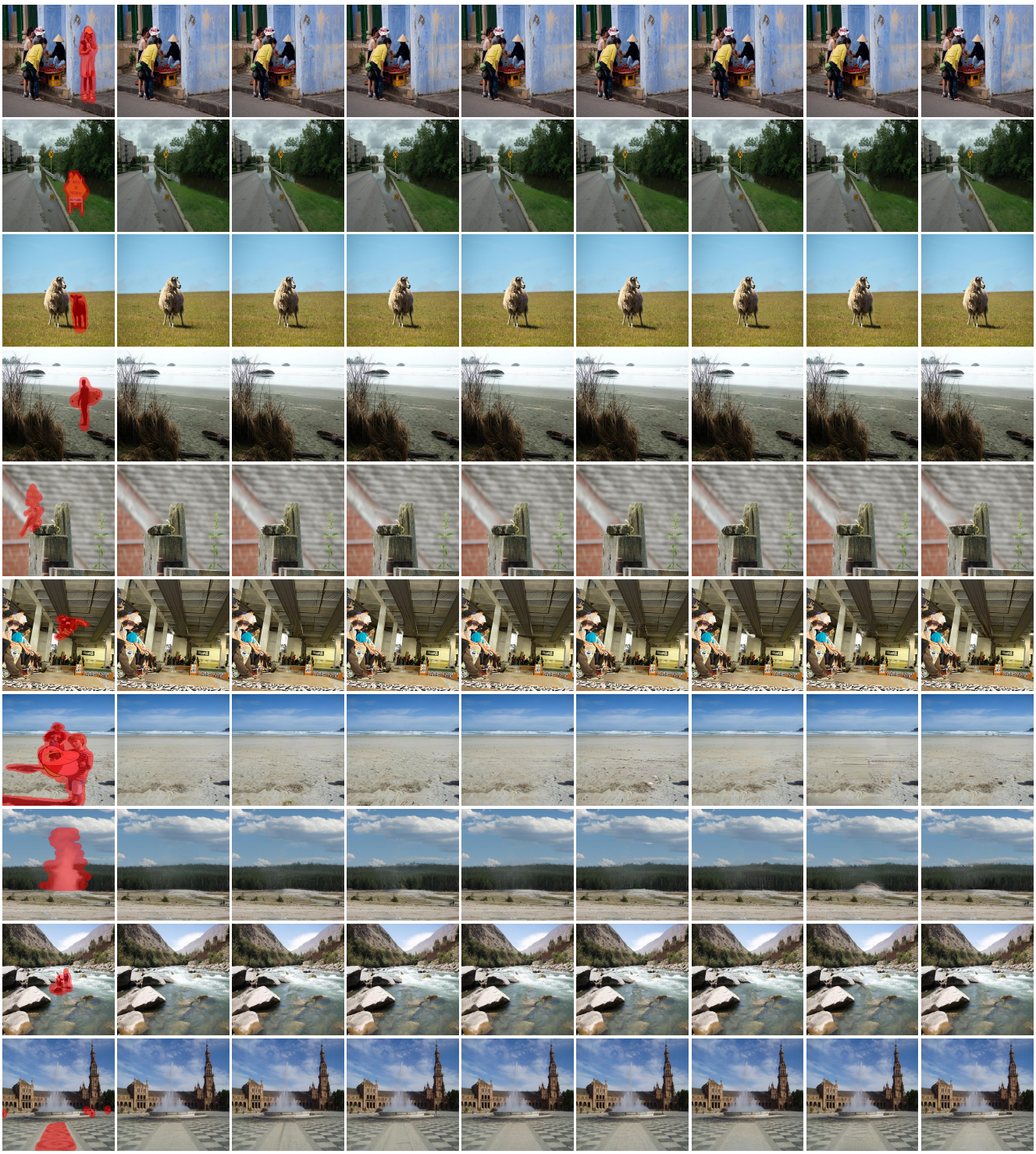


Figure S.7: Additional results of our 512 resolution model and other approaches using object masks.

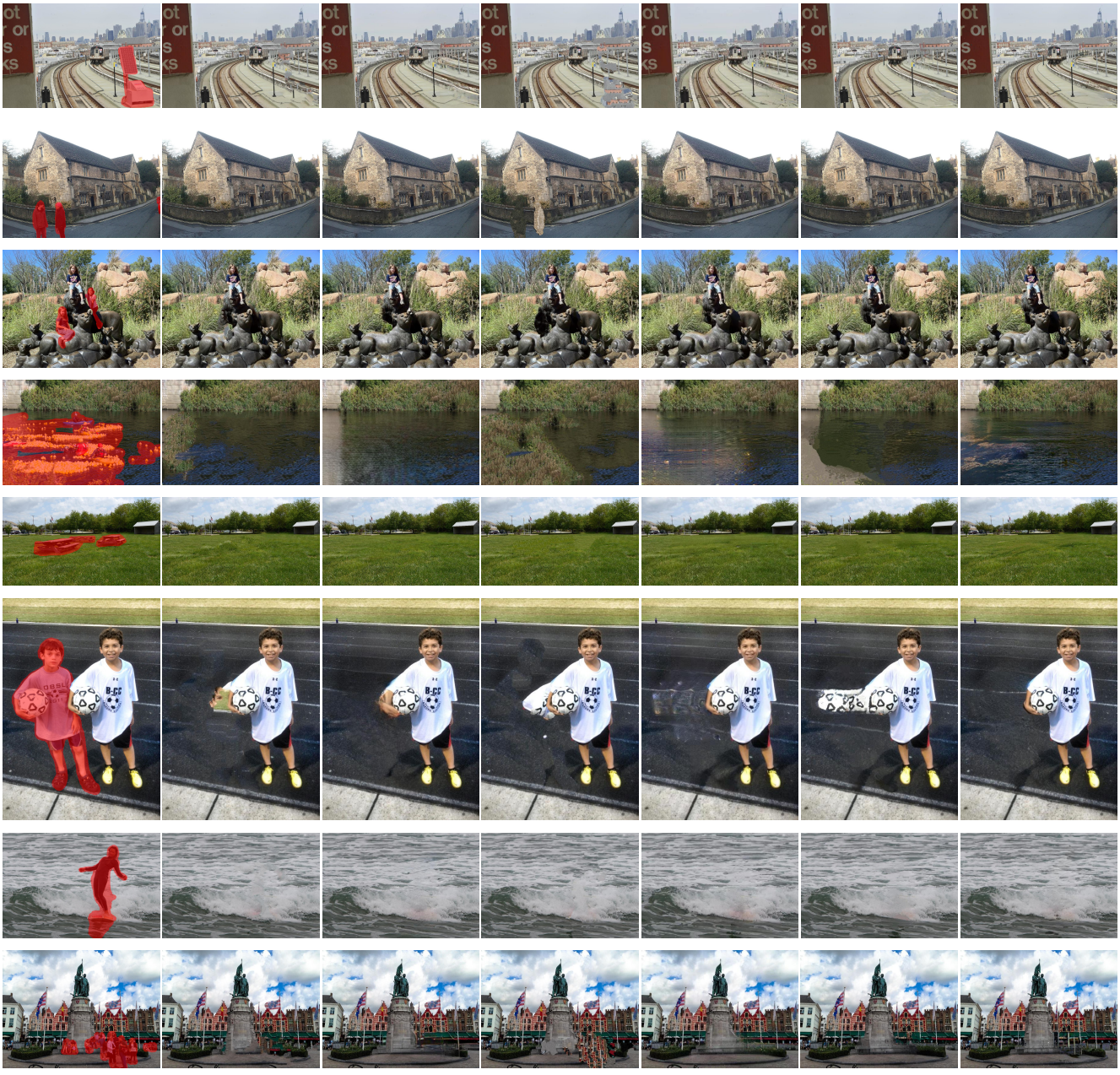


Figure S.8: More results from our user study. Zoom-in to compare the details.

Figure S.9: Failure cases of our approach.