MSG-GAN: Multi-Scale Gradients for Generative Adversarial Networks Supplementary Material

Animesh Karnewar
TomTom
animesh.karnewar@tomtom.com

Oliver Wang
Adobe Research
owang@adobe.com

Block	Operation	Act.	Output Shape
1.	Latent Vector	Norm	512 x 1 x 1
	Conv 4 x 4	LReLU	512 x 4 x 4
	Conv 3 x 3	LReLU	512 x 4 x 4
2.	Upsample	-	512 x 8 x 8
	Conv 3 x 3	LReLU	512 x 8 x 8
	Conv 3 x 3	LReLU	512 x 8 x 8
3.	Upsample	-	512 x 16 x 16
	Conv 3 x 3	LReLU	512 x 16 x 16
	Conv 3 x 3	LReLU	512 x 16 x 16
4.	Upsample	-	512 x 32 x 32
	Conv 3 x 3	LReLU	512 x 32 x 32
	Conv 3 x 3	LReLU	512 x 32 x 32
Model 1 ↑			
5.	Upsample	-	512 x 64 x 64
	Conv 3 x 3	LReLU	256 x 64 x 64
	Conv 3 x 3	LReLU	256 x 64 x 64
6.	Upsample	-	256 x 128 x 128
	Conv 3 x 3	LReLU	128 x 128 x 128
	Conv 3 x 3	LReLU	128 x 128 x 128
Model 2 ↑			
7.	Upsample	-	128 x 256 x 256
	Conv 3 x 3	LReLU	64 x 256 x 256
	Conv 3 x 3	LReLU	64 x 256 x 256
Model 3 ↑			
8.	Upsample	-	64 x 512 x 512
	Conv 3 x 3	LReLU	32 x 512 x 512
	Conv 3 x 3	LReLU	32 x 512 x 512
9.	Upsample	-	32 x 1024 x 1024
	Conv 3 x 3	LReLU	16 x 1024 x 1024
	Conv 3 x 3	LReLU	16 x 1024 x 1024
Model full ↑			

Table 1: Generator architecture for the MSG-ProGAN models used in training.

1. Architecture Details

MSG-ProGAN Tables 1 and 2 provide the detailed configurations of the generator and the discriminator of MSG-ProGAN respectively. After every block in the generator, a 1 x 1 conv layer is used to convert the output activation volume into an RGB image which is passed onto the discriminator. On the discriminator's side, these RGB images are combined with straight path activation volumes using the combine function ϕ. In case of ϕ_{simple}, a simple channelwise concatenation operation is used (see Table 2). For the ϕ_{lin_cat} variant of the combine function, a 1 x 1 conv layer is used to project the RGB images into activation space which is then followed by channelwise concatenation operation. The number of channels output by the 1 x 1 conv layer is equal to half of the output channels in that block of the discriminator, e.g. for block 3 (see Table 2), the output of the 1 x 1 conv layer is 32 x 256 x 256 and the output of ϕ_{lin_cat} operation is 96 x 256 x 256 (32 + 64). Finally, for the ϕ_{cat_lin}, the RGB images are first concatenated with the straight path activation volumes followed by a 1 x 1 conv layer. The number of channels output by this 1 x 1 conv layer is again equal to the prevalent number of channels in that block (e. g. 64 for block 3).

Model 1, *Model 2* and *Model 3* blocks of the generator (Tab 1) are used to synthesize 32 x 32, 128 x 128 and 256 x 256 sized images respectively. And, after every 3 x 3 conv operation the feature vectors are normalized according to the PixNorm [1] scheme (only for the generator).

MSG-StyleGAN The MSG-StyleGAN model uses all the modifications proposed by StyleGAN [2] to the ProGANs [1] architecture except the mixing regularization. Similar to MSG-ProGAN, we use a 1 x 1 conv layer to obtain the RGB images output from every block of the StyleGAN generator leaving everything else (mapping network, non-traditional input and style adaIN) untouched. The discriminator architecture is same as the ProGANs (and consequently MSG-ProGAN, Tab. 2) discriminator.

2. Additional Qualitative Results

Here we include additional results for further empirical validation. We show full resolution results from MSG-StyleGAN for the 256 x 256 Oxford102 flower dataset, and the MSG-ProGAN architecture for the 128 x 128 CelebA and LSUN bedroom datasets. The CelebA model was trained for $28M$ real images and obtained an FID of

Block	Operation	Act.	Output Shape
Model full ↓			
1.	Raw RGB images 0	-	3 x 1024 x 1024
	FromRGB 0	-	16 x 1024 x 1024
	MinBatchStd	-	17 x 1024 x 1024
	Conv 3 x 3	LReLU	16 x 1024 x 1024
	Conv 3 x 3	LReLU	32 x 1024 x 1024
	AvgPool	-	32 x 512 x 512
2.	Raw RGB images 1	-	3 x 512 x 512
	Concat/ϕ_{simple}	-	35 x 512 x 512
	MinBatchStd	-	36 x 512 x 512
	Conv 3 x 3	LReLU	32 x 512 x 512
	Conv 3 x 3	LReLU	64 x 512 x 512
	AvgPool	-	64 x 256 x 256
Model 3 ↓			
3.	Raw RGB images 2	-	3 x 256 x 256
	Concat/ϕ_{simple}	-	67 x 256 x 256
	MinBatchStd	-	68 x 256 x 256
	Conv 3 x 3	LReLU	64 x 256 x 256
	Conv 3 x 3	LReLU	128 x 256 x 256
	AvgPool	-	128 x 128 x 128
Model 2 ↓			
4.	Raw RGB images 3	-	3 x 128 x 128
	Concat/ϕ_{simple}	-	131 x 128 x 128
	MinBatchStd	-	132 x 128 x 128
	Conv 3 x 3	LReLU	128 x 128 x 128
	Conv 3 x 3	LReLU	256 x 128 x 128
	AvgPool	-	256 x 64 x 64
5.	Raw RGB images 4	-	3 x 64 x 64
	Concat/ϕ_{simple}	-	259 x 64 x 64
	MinBatchStd	-	260 x 64 x 64
	Conv 3 x 3	LReLU	256 x 64 x 64
	Conv 3 x 3	LReLU	512 x 64 x 64
	AvgPool	-	512 x 32 x 32
Model 1 ↓			
6.	Raw RGB images 5	-	3 x 32 x 32
	Concat/ϕ_{simple}	-	515 x 32 x 32
	MinBatchStd	-	516 x 32 x 32
	Conv 3 x 3	LReLU	512 x 32 x 32
	Conv 3 x 3	LReLU	512 x 32 x 32
	AvgPool	-	512 x 16 x 16
7.	Raw RGB images 6	-	3 x 16 x 16
	Concat/ϕ_{simple}	-	515 x 16 x 16
	MinBatchStd	-	516 x 16 x 16
	Conv 3 x 3	LReLU	512 x 16 x 16
	Conv 3 x 3	LReLU	512 x 16 x 16
	AvgPool	-	512 x 8 x 8
8.	Raw RGB images 7	-	3 x 8 x 8
	Concat/ϕ_{simple}	-	515 x 8 x 8
	MinBatchStd	-	516 x 8 x 8
	Conv 3 x 3	LReLU	512 x 8 x 8
	Conv 3 x 3	LReLU	512 x 8 x 8
	AvgPool	-	512 x 4 x 4
9.	Raw RGB images 7	-	3 x 4 x 4
	Concat/ϕ_{simple}	-	515 x 4 x 4
	MinBatchStd	-	516 x 4 x 4
	Conv 3 x 3	LReLU	512 x 4 x 4
	Conv 4 x 4	LReLU	512 x 1 x 1
	Fully Connected	Linear	1 x 1 x 1

Table 2: Discriminator Architecture for the MSG-ProGAN and MSG-StyleGAN Models used in training.

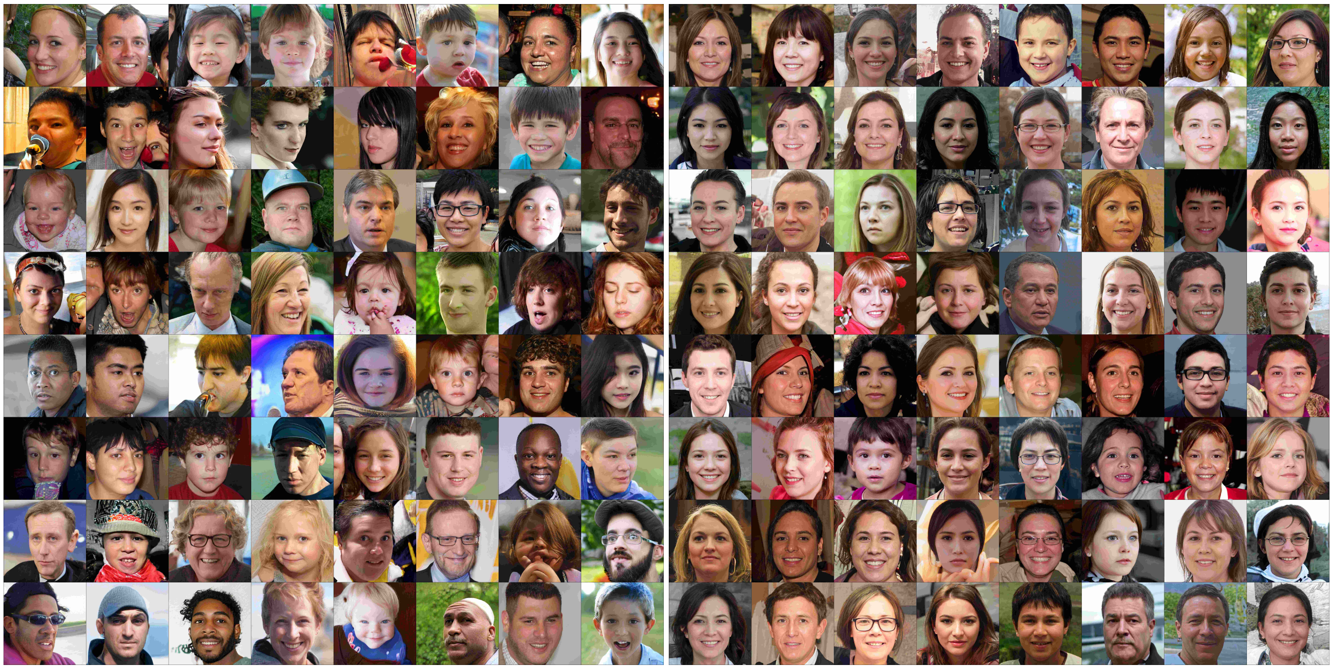
8.86. Because of the huge size of the LSUN bedrooms dataset (30M), we trained it for $150M$ real images (roughly 5 epochs) which resulted in an FID of 18.32. Figures 6 and 7 show the `128 x 128` (highest resolution) samples generated for the CelebA and LSUN bedrooms datasets respectively. Figure 4 and Fig 5 shows samples generated by the MSG-StyleGAN model at all resolutions on the Oxford Flowers and Cifar-10 datasets respectively. Figure 8 shows additional qualitative results (random samples) from the CelebA-HQ dataset, trained using our *Model full* architecture at `1024 x 1024` resolution.

3. Observations

In this section, we present some of our observations and hypotheses about the differences in results generated by our method and StyleGAN. We show an overview of randomly selected samples from both models in Fig 1. In our analysis of the results, we find that while the actual resulting image quality is very close, StyleGAN samples exhibit *slightly* higher variation in terms of pose. In contrast, MSG-StyleGAN results are *slightly* more globally consistent and more realistic. This trade-off between diversity and result quality is widely reported [3], and may explain some of the difference in FID score. Further investigation into methods to control either axis (realism vs diversity), and the impact this has on the FID score, would be an interesting avenue for future work.

We also conducted experiments investigating the role that the pixelwise noise added to each block of the StyleGAN generator plays in image generation. We found that on non-face datasets, these noise layers model *semantic* aspects of the images and not just stochastic variations, as was their initial intent [2] (see Fig 2). We observed that MSG-StyleGAN also shows this type of effect, although to a slightly less degree. We conjecture that this disentanglement between the stochastic and semantic features is more straightforward for the face modelling task (e.g., on CelebA-HQ and FFHQ datasets), and the different models sensitivity to this noise could contribute to some of the the performance differences we observe as well, on face vs non-face datasets.

As mentioned in the discussion section of the main paper, we do not use the mixing regularization technique described in the StyleGAN [2] work (the question of how to integrate such a regularization is an interesting direction for future work). However, we note that in spite of not using it, the model still learns to disentangle high level semantic features of the images due to the scale based constraint (see Fig. 3). As apparent from the figure, the high level mixing is much more coherent and generates more visually realistic results; while lower level mixing often generates incorrect visual cues, such as improper lighting and unbalanced hair. This shows that performance gains might be



(a) StyleGAN generated images

(b) MSG-StyleGAN generated images

Figure 1: Random generated samples for qualitative comparison between StyleGAN [2] and MSG-StyleGAN. All the samples were generated ***without*** truncating the input latent space for both because the FID calculation is done on non-truncated latent spaces. Best viewed zoomed in.

possible by ensuring proper style-based mixing at the low (coarse-grained) level of generation.

References

[1] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018. 1

[2] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, pages 4401–4410, 2019. 1, 2, 3, 5

[3] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *Advances in Neural Information Processing Systems*, 2017. 2

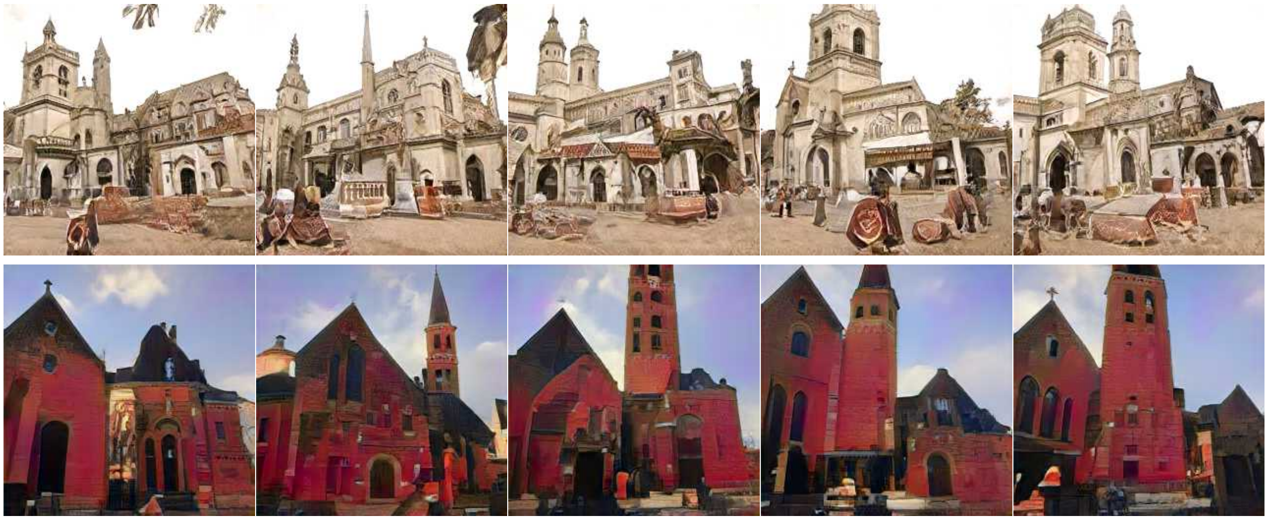


Figure 2: LSUN Church images generated by StyleGAN (top) and MSG-StyleGAN (bottom) using different realizations of the per-pixel noise while keeping the input latent vectors constant.

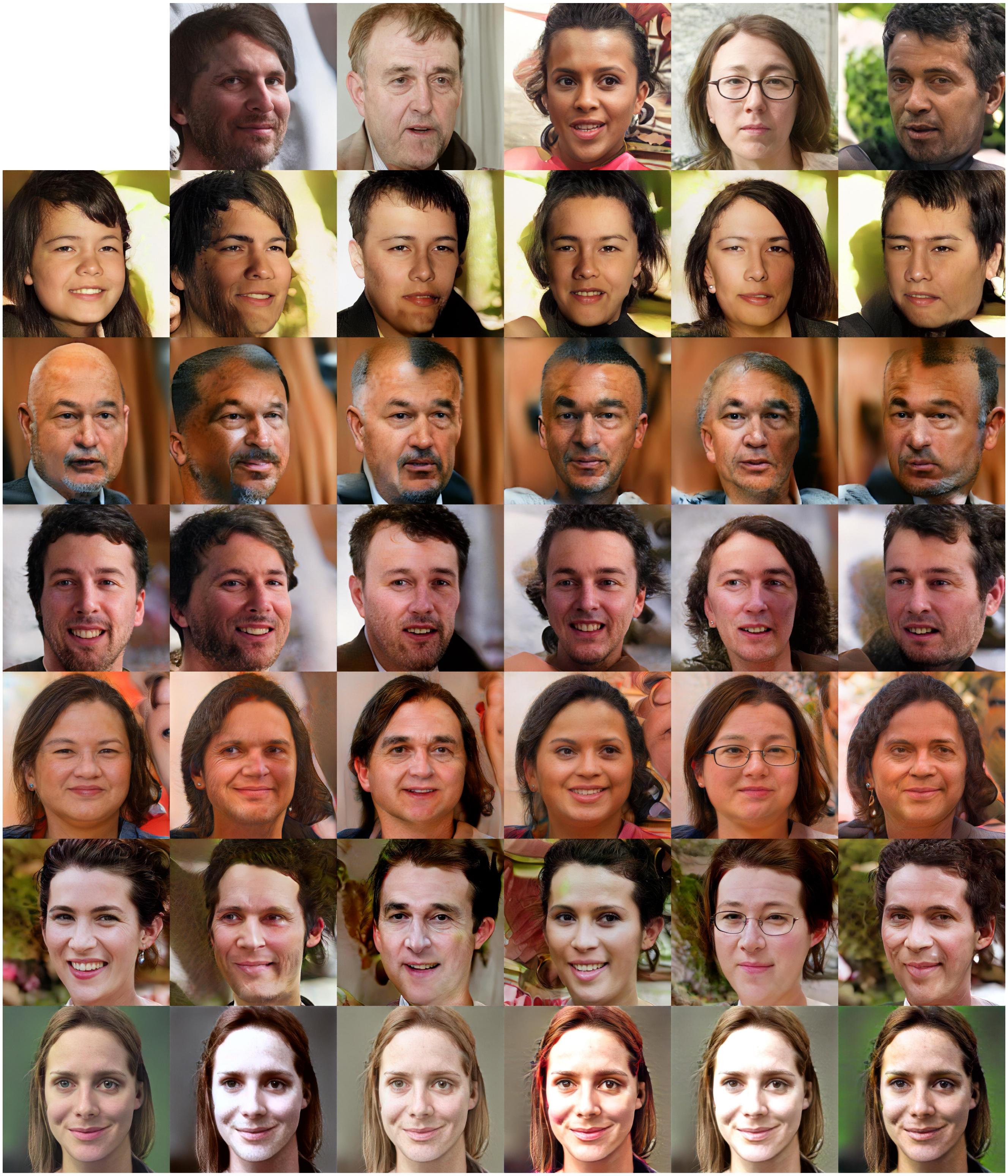


Figure 3: Images generated by mixing the styles coming from two different latent vectors at different levels (granularity) of generation. As in StyleGAN [2], the first column images are source 1 and first row are source 2. Rows numbered 2, 3, and 4 have the mixing at resolutions (`4 x 4` and `8 x 8`), while rows 5 and 6 at (`16 x 16` and 32 x 32), and the row 6 images are generated by swapping the source 2 latents at resolutions (`64 x 64` till `1024 x 1024`).

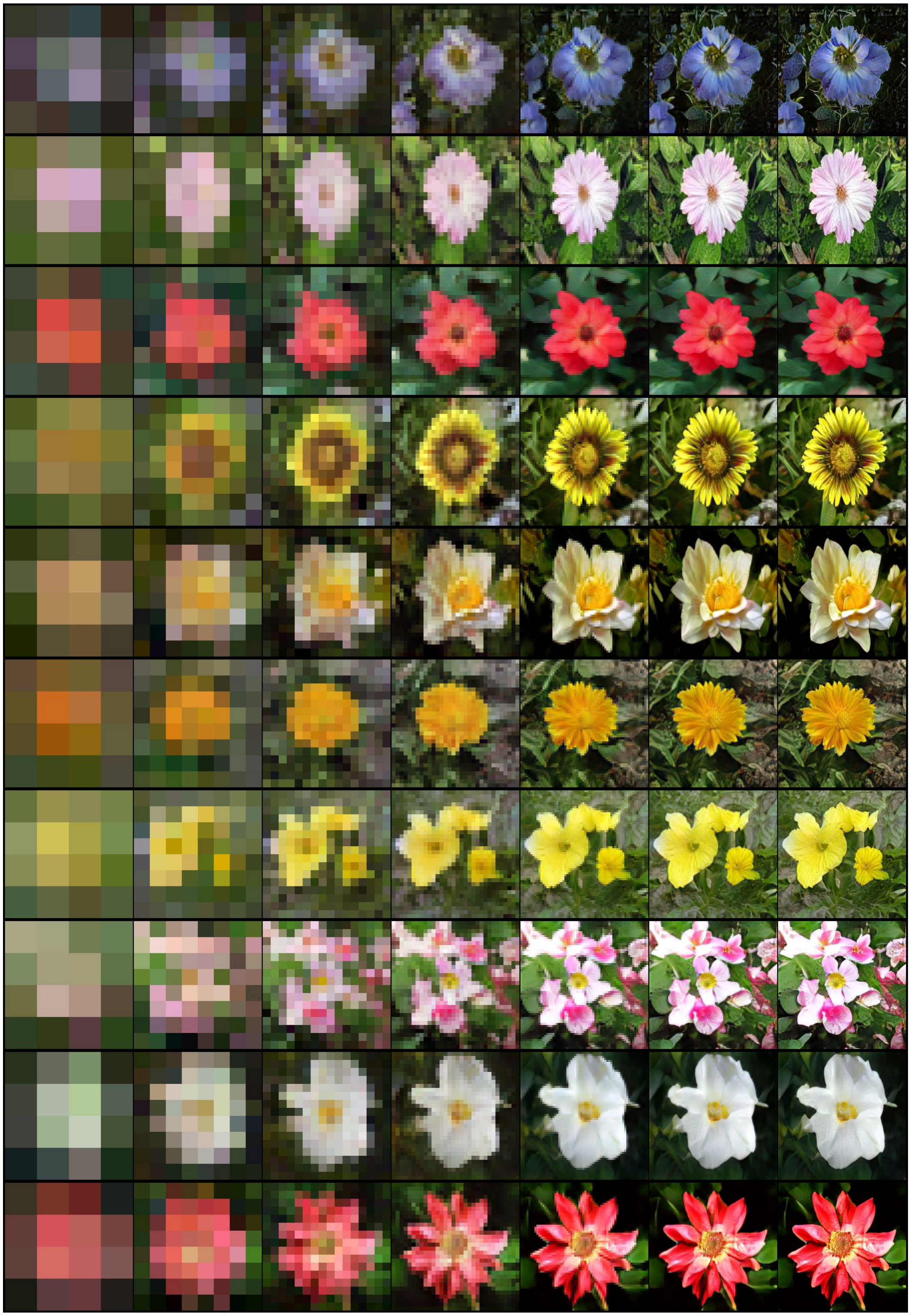


Figure 4: Random samples generated at all 7 resolutions for the Oxford102 flowers dataset.

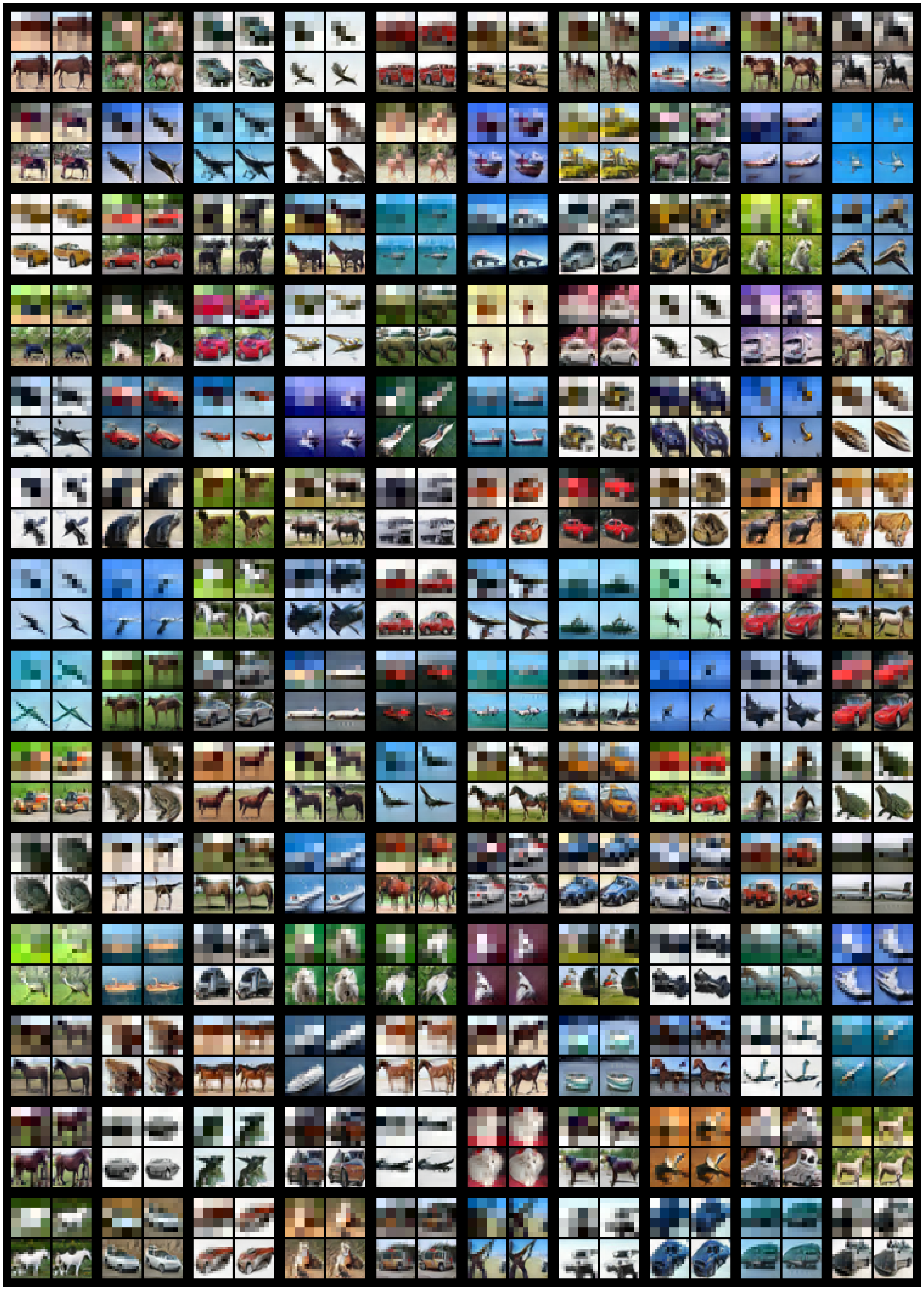


Figure 5: Random samples generated at all 4 resolutions for the CIFAR-10 dataset.

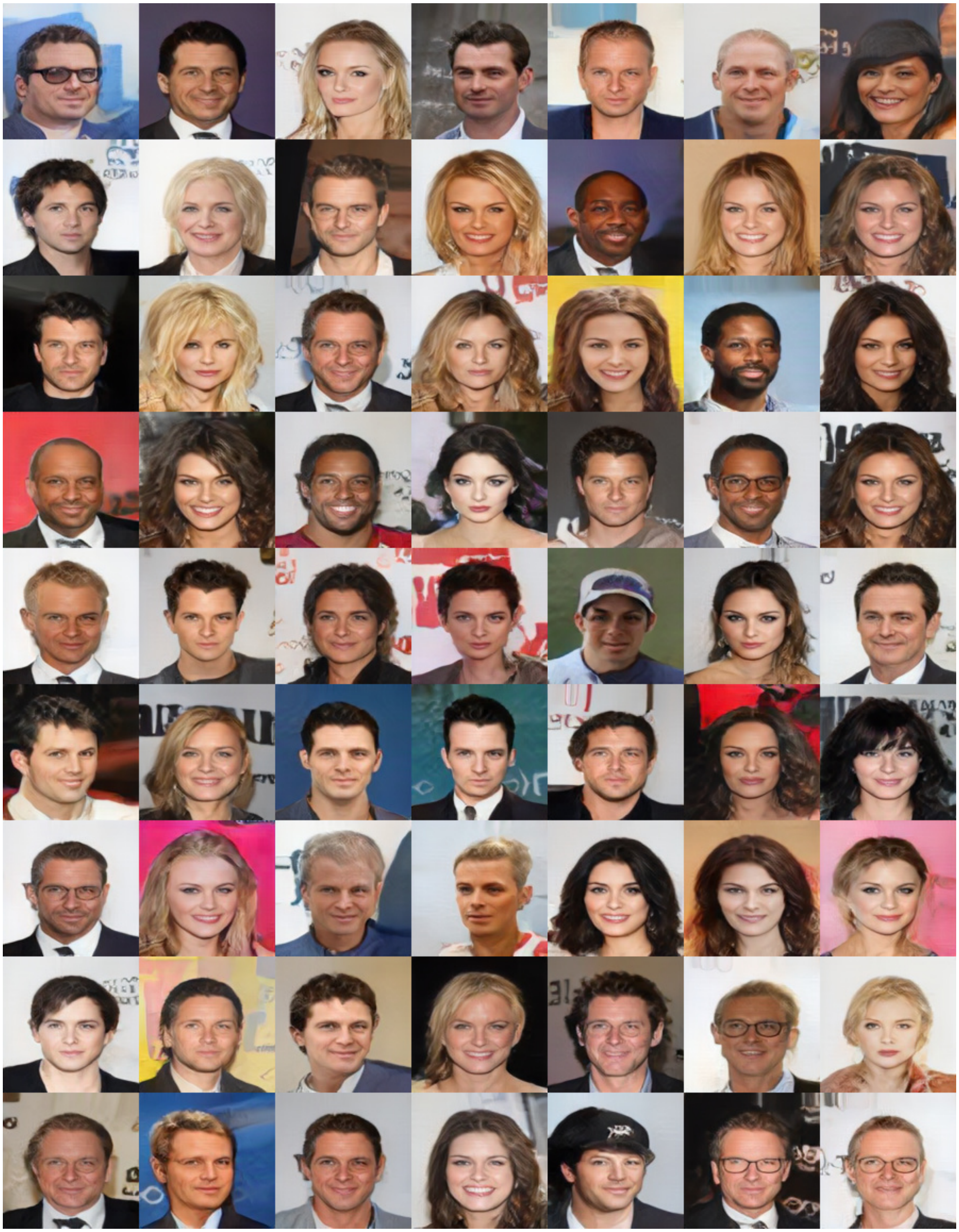


Figure 6: Random generated CelebA Faces at resolution `128 x 128`.

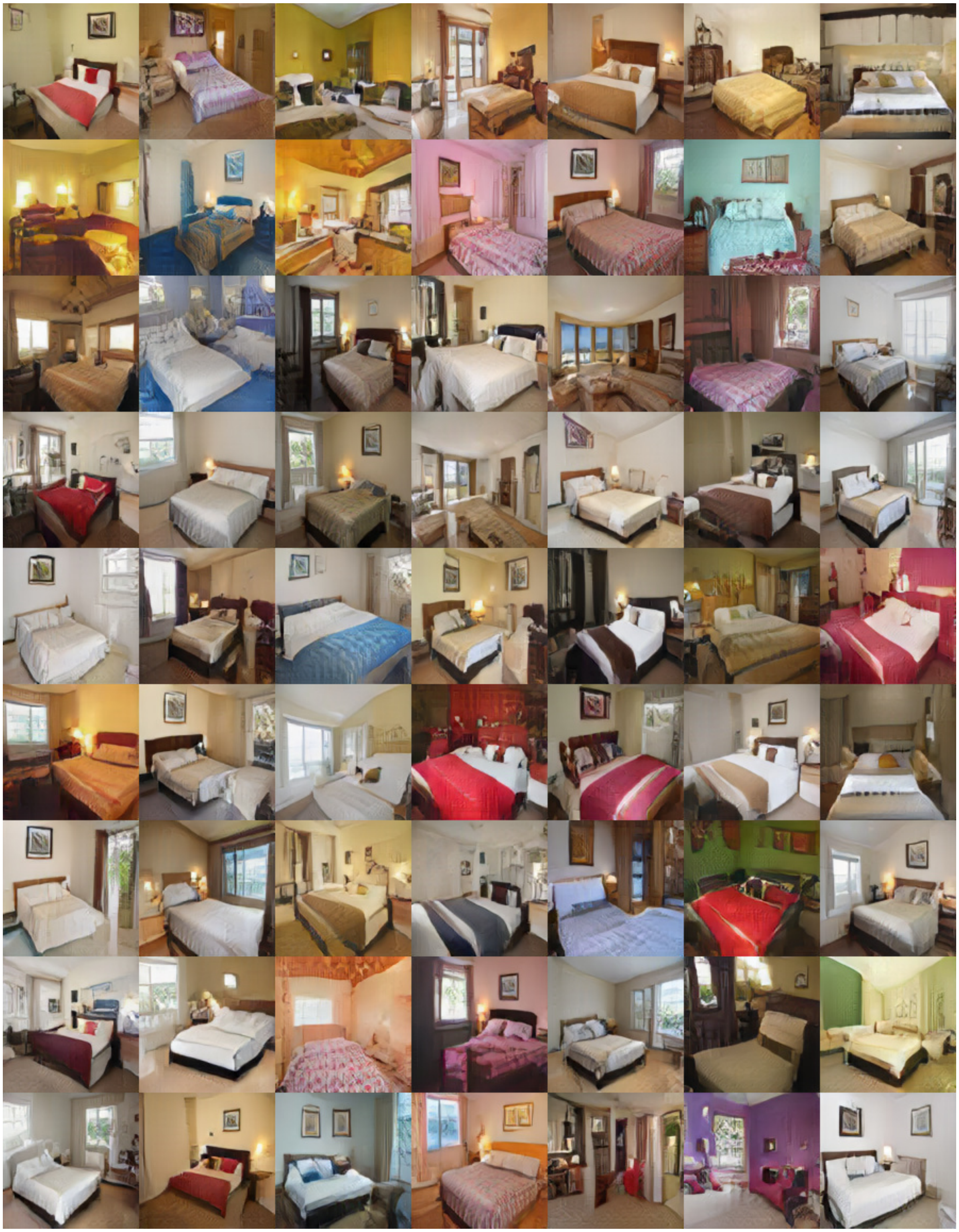


Figure 7: Random generated LSUN bedrooms at resolution `128 x 128`.

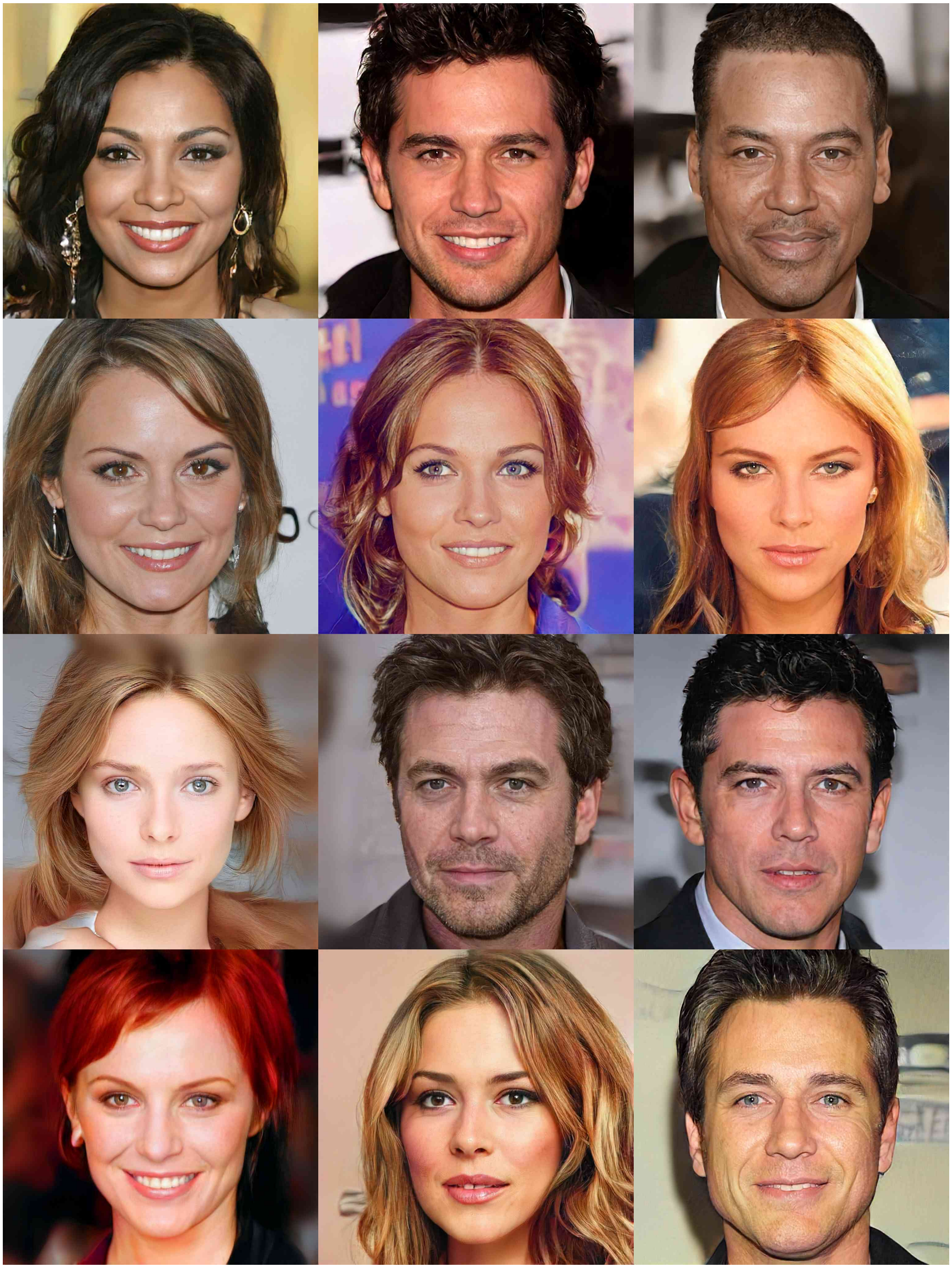


Figure 8: Random generated CelebA-HQ Faces at resolution `1024 x 1024`.