Image2StyleGAN: How to Embed Images Into the StyleGAN Latent Space? -Supplementary Material-

Rameen Abdal
KAUST
rameen.abdal@kaust.edu.sa

Yipeng Qin
KAUST
yipeng.qin@kaust.edu.sa

Peter Wonka
KAUST
pwonka@gmail.com

1. Additional Materials on Embedding

Dataset In order to test our embedding algorithm, we collect a small dataset of 25 images in five different categories: human faces, cats, dogs, cars and paintings (Figure 6).

Additional Embedding Results To further support our findings about the initial latent code in the main paper, we show more results in Figure 2. It can be observed that: for face images, initializing the optimization with the mean face latent code works better; while for non-face images, using the latent codes randomly sampled from a multivariate uniform distribution is a better option.

Quantitative Results on Defective Image Embedding Table 1 shows the corresponding quantitative results on defective image embedding (Figure 3 in the main paper). The results show that compared to non-defective faces, the embedded images of defective faces are farther from the mean face. This reaffirms that the valid faces form a cluster around the mean face.

Inherent Circular Artifacts of StyleGAN Interestingly, we observed that the StyleGAN model trained on the FFHQ dataset (officially released [2, 5]) inherently creates circular artifacts in the generated images, which are also observable in our embedding results (Figure 10). These artifacts are thus independent of our embedding algorithm and may be resolved by employing better pretrained models in the future.

Limitation of the ImageNet-based Perceptual loss All existing perceptual losses utilize the classifiers trained on the ImageNet dataset (*e.g.* VGG-16, VGG-19), which are restricted to the resolution of 224×224. While in our paper, we aim to embed images of high resolution (1024×1024) that are much larger than that of ImageNet images. Such inconsistency in the resolution may disable the learned image filters as they are scale-dependent. To this end, we follow the common practice [1, 3] and use a simple resizing trick to compute the perceptual loss on resized images of 256×256 resolution. As Figure 1 shows, the embedding results with the resizing trick outperform the ones at the original resolution. However, small details are lost during the resizing, which can slightly smoothen the embedding results. We expect to get better results with future perceptual losses that work on higher resolutions.

Figure 1: First column: original image (1024×1024). Second column: embedded image with the perceptual loss applied to resized images of 256×256 resolution. Third column: embedded image with the perceptual loss applied to the images at the original 1024×1024 resolution.

StyleGANs trained on Other Datasets To support our insights on the learned distribution, we further tested our embedding algorithm on the StyleGANs trained on three

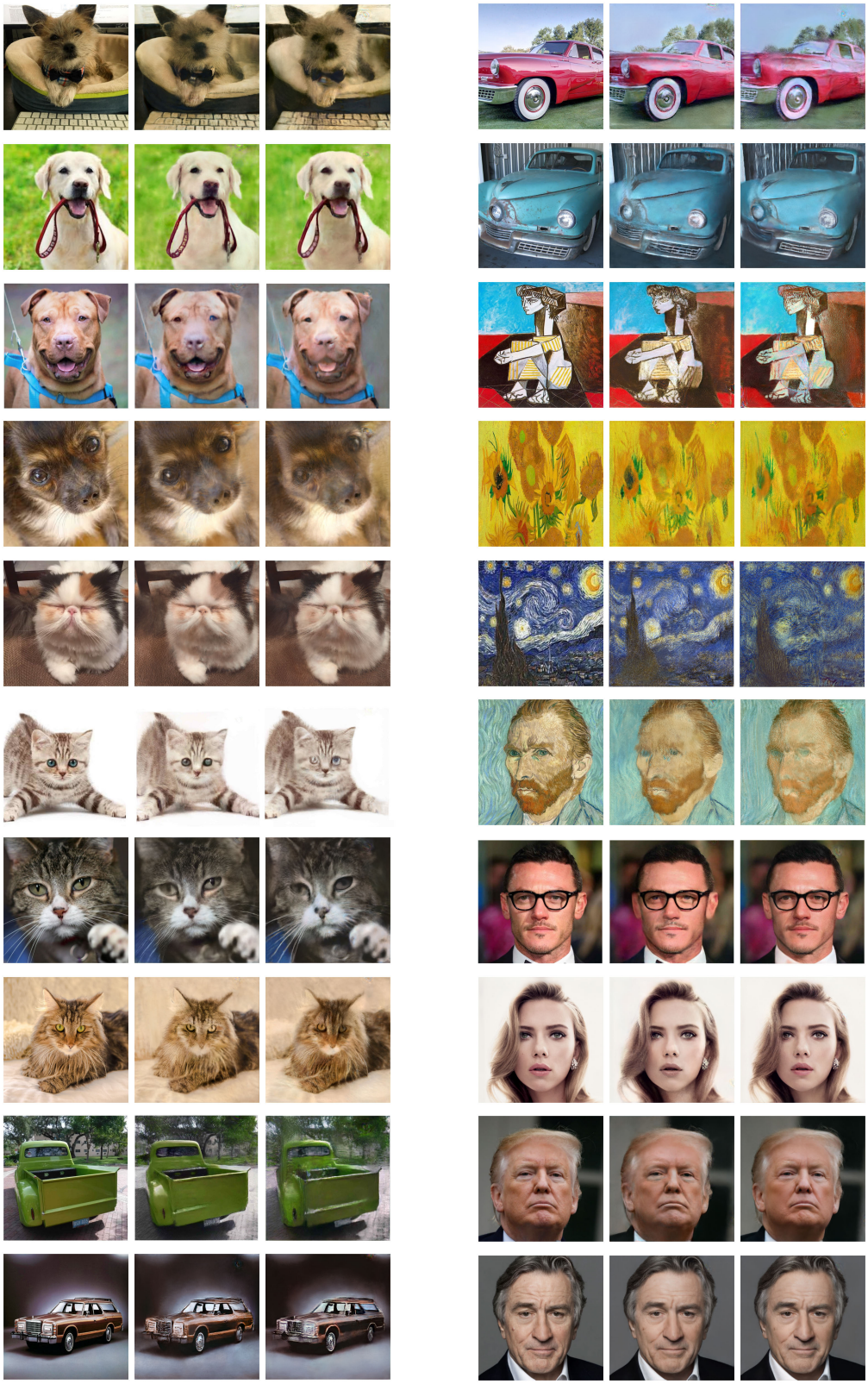
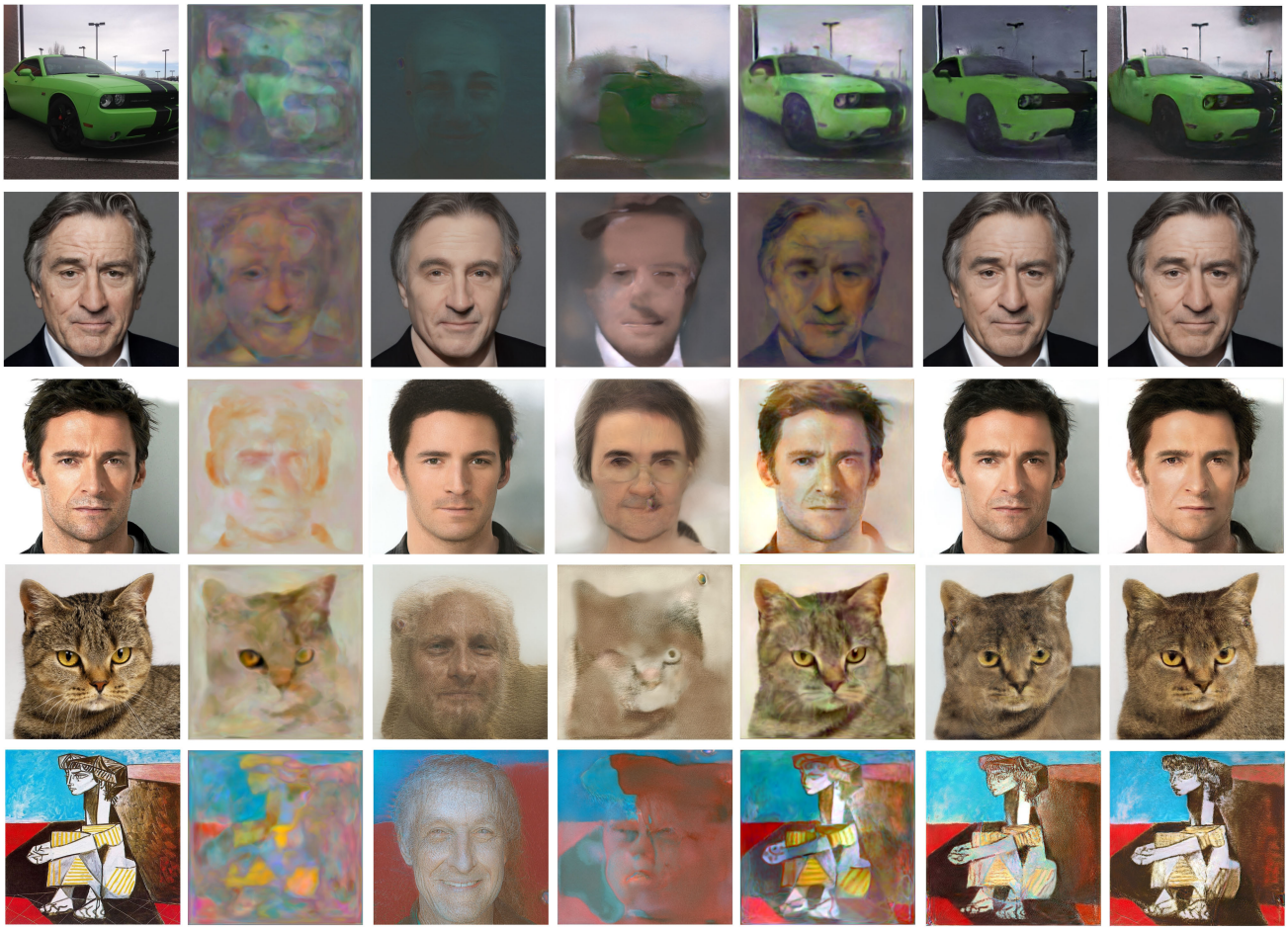


Figure 2: Additional Embedding Results into $W+$ space. Left column: the original images. Middle column: the embedded images with random latent code initialization. Right column: the embedded images with $\bar{w}$ latent code initialization.



(a) (b) (c) (d) (e) (f) (g)

Figure 3: Additional results on the justification of latent space choice.(a) Original images. Embedding results into the original space W: (b) using random weights in the network layers; (c) with $\bar{\mathbf{w}}$ initialization; (d) with random initialization. Embedding results into the $W+$ space: (e) using random weights in the network layers; (f) with $\bar{\mathbf{w}}$ initialization; (g) with random initialization.

Defect	$L(\times 10^5)$	$\|w^* - \bar{\mathbf{w}}\|$
non-defective	0.204	29.19
Eyes	0.271	34.90
Nose	0.311	39.20
Mouth	0.301	37.04
Eyes and Mouth	0.233	39.62
Eyes, Nose and Mouth	0.285	37.59

Table 1: Quantitative results on defective image embedding (Figure 3 in the main paper). L is the loss after optimization. $\|w^* - \bar{\mathbf{w}}\|$ is the distance between the latent codes w^* and $\bar{\mathbf{w}}$ of the average face.

more datasets: the LSUN-Car (512×384), LSUN-Cat (256×256) and LSUN-Bedroom (256×256) datasets. The embedding results are shown in Figure 7. It can be observed that the quality of the embedding is poor compared to that of the StyleGAN trained on the FFHQ dataset. The linear interpolation (image morphing) results of LSUN-Cat, LSUN-Car, and LSUN-Bedroom StyleGANs are shown in Figure 8 (a), (b) and (c) respectively. Interestingly, we observed that linear interpolation fails on the LSUN-Cat and LSUN-Car StyleGANs. Recall that the FFHQ human face dataset is of very high quality in terms of scale, alignment, color, poses *etc*., we believe that the low quality of the LSUN datasets is the source of such failure. In other words, the quality of the data distribution is one of the key components to learn a meaningful model distribution.

Figure 4: Additional morphing results between two embedded images (the left-most and right-most ones).

Figure 5: Image embedding using different constant noises.

Additional Results on the Justification of Latent Space Choice Figure 3 shows additional results (cat, dog, car) on the justification of our choice of latent space W^+. Similar to the main paper, we can observe that: (i) embedding into W directly does not give reasonable results; (ii) the learned network weights is important to good embeddings.

Clustering or Scattering? To support our insight that only face images form a cluster in the latent space, we compute the $L2$ distances between the embeddings of all pairs of test images (Figure 9). It can be observed that the distances between the faces are relatively smaller than those of other classes, which justifies that they are close to each other in the W^+ space and form a cluster. For images in other classes, especially the paintings, the pairwise distances are much higher. This implies that they are scattered in the latent space.

Justification of Loss Function Choice Figure 11 validates the algorithmic choice of the loss function used in the main paper. It can be observed that (i) matching the image features at multiple layers of the VGG-16 network works better than at a single layer; (ii) the combination of pixel-wise MSE loss and perceptual loss works the best.

Influence of Noise Channels Figure 5 shows that restarting the embedding with a different noise leads to similar results. In addition, we observed significantly worse quality when resampling the noise during the embedding (at each update step). To this end, we kept the noise channel constant during the embedding for all our experiments.

2. Additional Results on Applications

Figure 4 shows additional results of the image morphing. Figure 12 shows the complete table of the style transfer results between different classes. The results support our insight that the multi-class embedding works by using an underlying human face structure (encoded in the first couple of layers) and painting powerful styles onto it (encoded in the latter layers). Figure 14 shows additional results on the expression transfer. We also include an accompanying video in the supplementary material to show it works with noisy images taken by a commodity camera in a typical office environment. The random walk results (of two classes 'human faces' and 'cars') from the embedded image towards the mean face image are also shown in videos.

References

[1] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 1

[2] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018. 1

[3] Samuli Laine. Feature-based metrics for exploring the latent space of generative models, 2018. 1

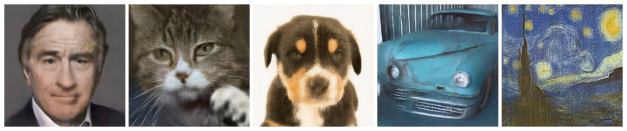
[4] Jesús P Mena-Chalco, Luiz Velho, and RM Cesar Junior. 3d human face reconstruction using principal components spaces. In *Proceedings of WTD SIBGRAPI Conference on Graphics, Patterns and Images*, 2011. 13

[5] Timo Aila Tero Karras, Samuli Laine. Stylegan - official tensorflow implementation. https://github.com/NVlabs/stylegan, 2018. 1

Figure 6: The collected 25 images of our dataset. First row: human faces. Second row: cats. Third row: dogs. Fourth row: cars. Fifth row: paintings.

(a)

(b)



(c)

Figure 7: Embedding results of StyleGANs trained on (a) LSUN-Car, (b) LSUN-Cat and (c) LSUN-Bedroom datasets. For each subfigure, the first row shows the embedding results of the images in 5 different classes in our dataset. The second row shows the embedding results of the images of the corresponding class in our dataset ("cars" in (a) and "cats" in (b)). Note that (c) has only one row because we did not collect bedroom images in our dataset.

(a)

(b)

(c)

Figure 8: Results on linear interpolations (image morphing) in the latent spaces of StyleGANs trained on (a) LSUN-Cat (b) LSUN-Car (c) LSUN-Bedroom datasets.

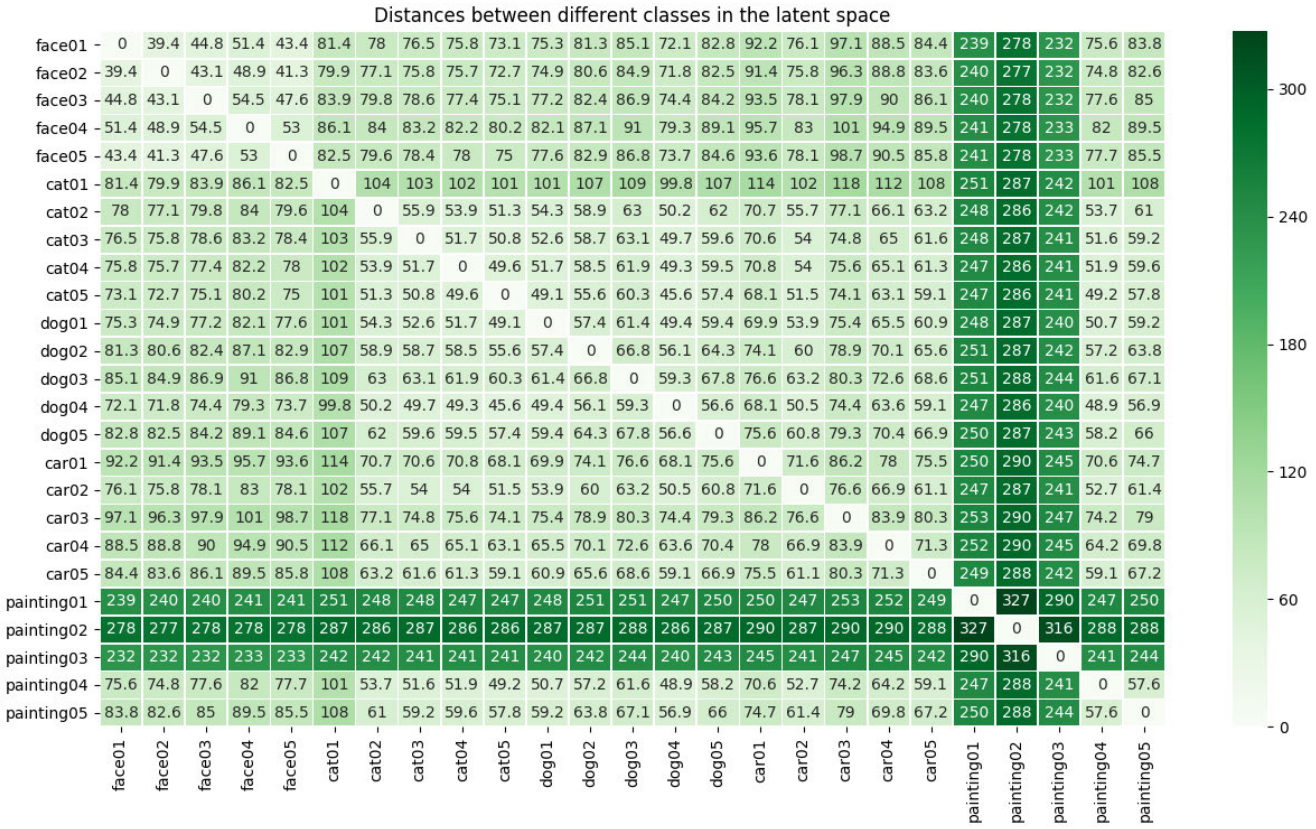


Figure 9: Heat map of the inter- and intra-class $L2$ distances between embedded images.

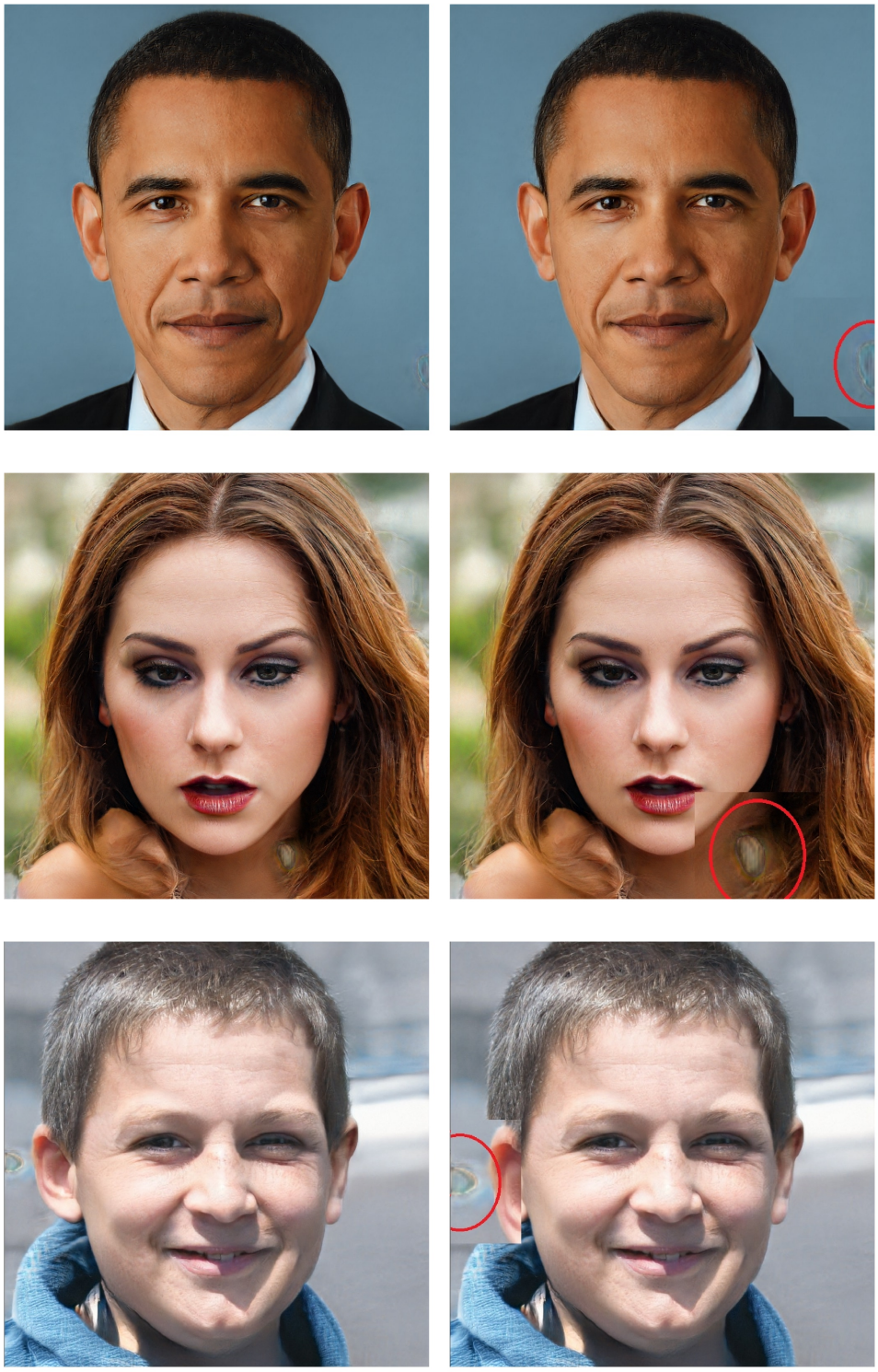


Figure 10: Inherent circular artifacts of StyleGAN. First row: circular artifacts in the embeded images. Second and third rows: randomly generated images. Left column: images with circular artifacts. Right column: highlighted artifacts by zooming in their local neighbourhood.

Figure 11: Additional results of the algorithmic choice justification on the loss function. Each row shows the results of an image from the five different classes in our test dataset respectively. From left to right, each column shows: (1) the original image; (2) pixel-wise MSE loss only; (3) perceptual loss on VGG-16 $conv3\_2$ layer only; (4) pixel-wise MSE loss and VGG-16 $conv3\_2$; (5) perceptual loss only; (6) our loss function .

Figure 12: Complete table of the style transfer results. Left-most column: the embedded style image. First row: the embedded content images.

(a)

(b)

Figure 14: Additional results on expression transfer. In each subfigure, the first row shows the reference images from IMPA-FACES3D [4] dataset; in the following rows, the middle image in each of the examples is the embedded image, whose expression is gradually transferred to the reference expression (on the right) and the opposite direction (on the left) respectively.