VIGFace: Virtual Identity Generation for Privacy-Free Face Recognition dataset

Supplementary Material

Face Recognition Model Training Configurations	
Head	AdaFace
Margin (m)	0.4
Scale (s)	64
Augmentation	Random Erase × Rescale × Jitter
Augmentation ratio	$0.2 \times 0.2 \times 0.2$
Reduce LR epochs (S, L)	[24, 30, 36], [12, 20, 24]
Epochs (S, L)	40, 26
Backbone	IR-SE50
Batch Size	512
Initial Learning Rate	0.1
Weight Decay	5e-4
Momentum	0.9
FP16	True
Optimizer	SGD

Table 3. Configurations for training the FR network. (S) and (L) represent training using the small dataset (< 1.0M) and the large dataset ($\geqq 1.0$M), respectively.

A. Implementation Details

For stage 1: We use a modified ResNet-50 [9] and ArcFace [9] to train the face recognition (FR) backbone. The CASIA-WebFace [22] dataset serves as training data. Following the alignment method in [32], the facial images are aligned with a resolution of 112×112. After alignment, the pixel values are normalized, with both the mean and standard deviation set to 0.5. We set the mini-batch size for the real dataset to 512 and use Stochastic Gradient Descent (SGD) as the optimizer, with a weight decay of 5e-4 and momentum of 0.9. The initial learning rate is set to 0.1 and divided by 0.1 in the 24th, 30th, and 36th epochs, with training concluding at the 40th epoch. The ArcFace hyperparameters for the margin m and the scale factor s are set to 0.5 and 30, respectively.

For stage 2: To generate synthetic face images, we explore the best setting for the diffusion model. To facilitate comparison, we utilized the widely adopted DiT-B model for all experiments. Since the traditional FR model typically employs a resolution of 112×112 for face images, we set the window size of the DiT patch extractor to 4. We follow the publicly released implementation of DDIM [46] with the cosine noise scheduler [7]. Following the approach of the previous method [31], we employ the enforced zero terminal SNR and trailing timesteps. The diffusion model is trained for 5M iterations with a batch size of 512 using AdamW Optimizer [30, 33] with a learning rate of 1e-4. For sampling, we used classifier-free guidance implementation [19] in 50 time steps.

Implementation details for FR training: We provide the configurations for the FR network training that are used in Sec. 4.2. We strongly refer to the training methods proposed in [15, 28, 55] to set the hyperparameters. Detailed configurations can be found in Tab. 3. We adjusted training epochs and learning rate scheduling strategies based on the dataset size. Following [28], we employ random erasing, rescaling, and color jittering as data augmentation, especially when training with AdaFace. Random erasing is applied by filling randomly selected regions with pixel values of 0. The erased region size is randomly set between 0.02 and 0.33 times the original image width, with an aspect ratio varying between 0.3 and 3.3. For rescaling, each face image is first shrunk and then restored to its original dimensions. The shrinking ratio is randomly selected between 0.2 and 1.0 times the original image width. To ensure diversity, we randomly apply one of the following interpolation methods during shrinking and restoration: nearest-neighbor interpolation, bilinear interpolation, bicubic interpolation, pixel area relation interpolation, and Lanczos4 interpolation. For color jittering, we randomly adjust the brightness, contrast, and saturation of the input image, with each factor modified within a range of 0 to 0.5 relative to the original image.

B. FR training using unified configuration

Training configuration, such as training loss, augmentations, and hyperparameters, is vital to the performance of FR models. However, conventional works have conducted the training based on their own configuration. In this section, we present reproduced experiments trained under the unified implementation except the training dataset in Tab. 4. As the training code of conventional methods for SFR is not released, we utilize margin(m) with 0.5, and scaling factor(s) with 30 and 64, following the original paper [9] settings.

As shown in Fig. 7, we observed that ArcFace with a scaling factor (s) of 64 failed to train stably in the conventional datasets due to the gradient explosion. The high s makes the softmax operation more steep near the decision boundary [28]. This leads to a gradient explosion during training when the data includes unrecognizable samples or label noise. CASIA-WebFace is known to include around $9.3\% - 13.0\%$ of label noise data [52]. In contrast, since VIGFace has high consistency, it is free from label noise and enables stable backbone training.

Note that the entire framework in VIGFace was trained using only CASIA-WebFace, without any external datasets

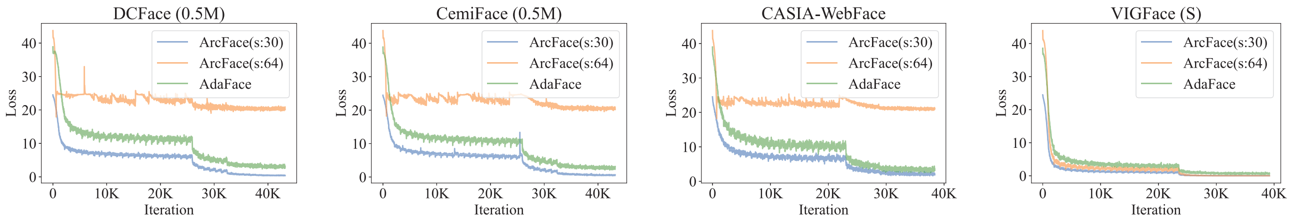


Figure 7. Loss log during training of FR backbone.

Training Dataset	Method	Average Accuracy	IJB-C [34] TPR@FPR 1e-4	1e-3	1e-2
CASIA-Webface	Arc. (s:30)	94.28	83.44	91.22	96.22
	Arc. (s:64)	70.16	15.07	29.26	50.30
	Ada.	94.93	77.65	92.66	96.89
DCFace (0.5M)	Arc. (s:30)	84.80	60.29	76.42	88.97
	Arc. (s:64)	71.50	11.37	22.62	41.07
	Ada.	89.07	77.24	87.51	94.01
CemiFace (0.5M)	Arc. (s:30)	85.57	37.85	77.33	90.09
	Arc. (s:64)	72.88	18.92	33.81	54.94
	Ada.	90.54	83.11	90.41	95.39
HSFace10K	Arc. (s:30)	85.39	73.94	82.18	89.67
	Arc. (s:64)	71.22	14.16	27.24	45.81
	Ada.	90.22	86.10	91.04	94.92
VIGFace (S)	Arc. (s:30)	91.19	69.27	82.64	91.78
	Arc. (s:64)	92.37	72.53	85.14	92.98
	Ada.	92.56	80.00	89.22	94.99
VIGFace (B)	Arc. (s:30)	94.00	80.33	88.87	94.69
	Arc. (s:64)	94.27	81.69	89.79	95.15
	Ada.	94.64	83.29	91.22	95.97

Table 4. Re-implemented FR benchmark results trained under unified configurations. The FR models used in the table were trained with the same implementation details, following original papers [9, 28].

or a pre-trained large model such as CLIP [41].

C. Stage 1 : Virtual prototype assignment

Performance of backbone in Stage 1: We evaluate the performance of the trained backbone achieved at the end of stage 1. Given that our approaches employ virtual prototypes, it is essential to guarantee that the virtual prototype does not affect the FR backbone training. As can be seen in Tab. 5, our approach maintains the benchmark face recognition performance with negligible changes.

Pre-assigning vs Sampling: In Stage 1, VIGFace employs a pre-assignment strategy for virtual prototypes within the feature space, maximizing inter-prototype distances through the ArcFace loss. The ArcFace loss explicitly enforces a substantial margin between distinct identities in the feature space, ensuring that virtual prototypes are highly discriminative. This potentially yields superior identity separation compared to random sampling approaches, which rely on fixed similarity thresholds. Our theoretical analysis for the high separation of virtual identities is also supported by our property analysis in Fig. 6a. In addition, since the embedding vectors used in the generative model are closely aligned, VIGFace produces synthetic faces that follow the actual data distribution better than the random sampling method.

Dataset	Real Prototype	Virtual Prototype	Average Accuracy	IJB-C
CASIA-Webface	10.5K	-	94.28	83.44
		60K	94.30	83.46

Table 5. Benchmark results of the stage 1: FR model which employs different number of virtual prototypes. For IJB-C, TPR@FPR=1e-4 is reported.

Although training virtual prototypes for a huge number of identities (*e.g.* millions or billions) requires a large-scale classifier, making VIGFace computationally intensive compared to sampling methods. However, this challenge can be mitigated by integrating techniques such as Partial-FC [1], which optimizes computational efficiency while maintaining performance.

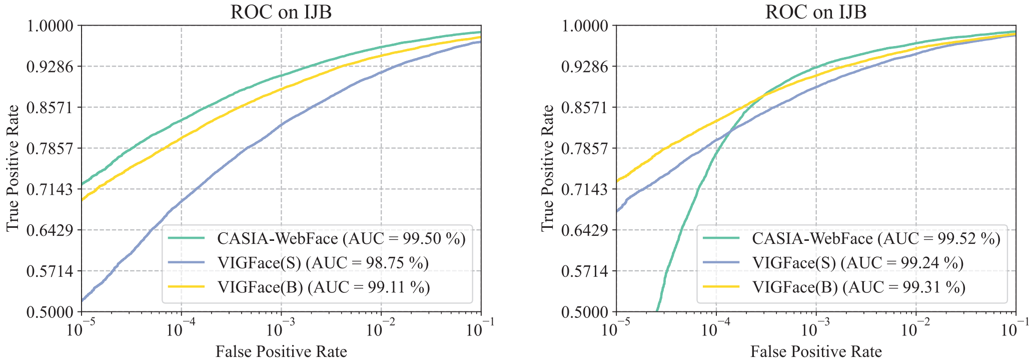
Scalability of virtual prototype: Our method demonstrates that all virtual prototypes can nearly maintain orthogonality after stage 1. From probabilistic geometry [6], if there are N identities following a uniform distribution, the minimum cosine distance between two vectors in a feature space of dimension d can be approximated as follows:

$$\cos(\theta) \approx \sqrt{\frac{\log N}{d}}. \tag{12}$$

This implies that the 512 feature dimension, which is used in our experiments, is large enough to sort 10^8 identity features while preserving nearly orthogonal.

D. ROC of IJB-C benchmarks

We compare the Receiver Operating Characteristic (ROC) curves using the IJB-C [34] dataset. Fig. 8 presents the ROC curves of IJB-C for VIGFace. We observe that VIGFace



(a) IR-SE50 + ArcFace (s:30).

(b) IR-SE50 + AdaFace.

Figure 8. ROC on IJB-C benchmarks.

outperforms real datasets at low FPR ($< 1e{-}4$) on AdaFace trained models.

E. Face Image Quality distribution

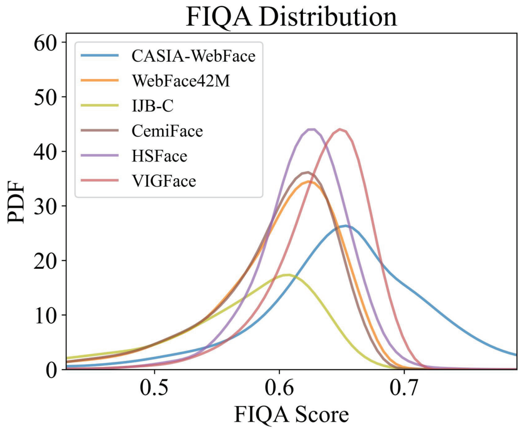


Figure 9. FIQA score distribution of various methods. For better visibility, scores are min-max normalized.

We provide the Face Image Quality (FIQ) score distribution of VIGFace and conventional methods obtained using a SOTA face image quality assessment method [4]. Fig. 9 shows FIQA score distribution of various synthetic datasets. To enhance clarity and facilitate comparison, the figure presents the normalized values. Note that a higher FIQ score does not indicate a good dataset for FR training. Ideally, FR training requires a diverse dataset, ranging from easy to hard, for optimal performance. As can be seen in Fig. 9, HSFace and CemiFace considerably include hard-case images. Since the IJB set also consists of mixed-quality images, training with these datasets might be beneficial for the IJB benchmark. However, samples in Fig. 18 - Fig. 20 show that HSFace and CemiFace contain significant low-quality images, such as blurred or distorted faces.

F. Similarity distribution analysis

We provide the similarity distribution of various datasets, including real and synthetic datasets. To achieve embedding features, we utilized a pre-trained ArcFace model trained on the Glint-360K dataset. As shown in Fig. 10, CASIA-WebFace contains some label-noised samples that can hinder stable backbone training. In other datasets, we observe significant overlap between positive and negative distributions. This implies that the dataset contains ID-flipped samples (or may be impossible to recognize), which can also make backbone training unstable. Meanwhile, VIGFace demonstrates high consistency and separation in the dataset.

G. Samples using multi-view landmark

We generate face images using landmark images with various 25 poses. The samples generated from both real and virtual ID prototypes are presented in Figs. 12 to 17. In the figure, we report the cosine similarity between the class center $\overline{f_k}$ and the generated image x_k. As shown in the figure, VIGFace can generate pose variational images without identity flipping, maintaining high similarity.

H. Samples from failure case

In this section, we compare the failure case in terms of three important properties of the datasets, which are consistency, separability, and diversity. We sample images from the lowest 5% subjects of each property to illustrate the effectiveness of VIGFace in the failure case.

Subjects exhibit the lowest consistency Fig. 18 shows the failure case images generated by conventional methods [2, 5, 29, 35, 40, 48, 55] and our methods. We report virtual subjects that exhibit the lowest 5% class consis-

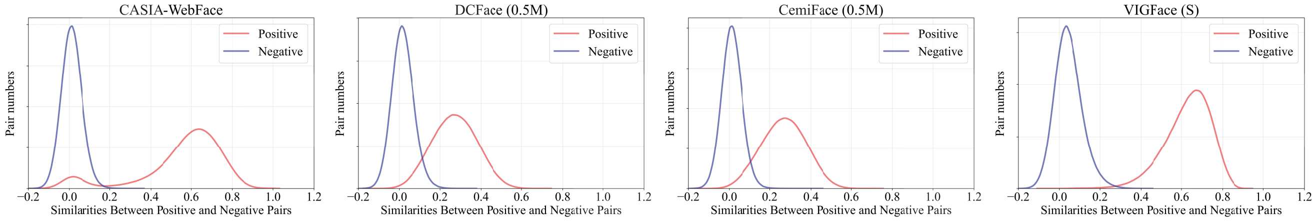


Figure 10. Similarity Distribution of various dataset.

tency in each dataset. In other words, the depicted subjects have a propensity to label-flip bias. As illustrated in the figure, VIGFace demonstrates outstanding consistency while maintaining high diversity in image conditions. This suggests a minimal risk of label-flip bias and supports effective FR training. DigiFace also maintains high consistency, as it is based on 3D modeling. However, since the appearance of the DigiFace dataset does not align with real face images, the performance of an FR backbone trained on DigiFace is considerably subpar for real-world use cases.

Subjects exhibit the lowest separability Fig. 19 shows the failure case images generated by conventional [2, 5, 29, 35, 40, 48, 55] and our methods. We report virtual subjects among those who exhibit the highest 5% cosine similarity. Generating overly similar or identical objects can result in label-noised data and negatively impact FR training. As shown in the figure, conventional methods often generate output that resembles the same object. This indicates a lack of the ability to generate entirely novel individuals. For example, SynFace samples exhibit high similarities between objects because they utilize a mix-up to generate face images. As GANDiffFace relies on StyleGAN to create virtual identities, it shows limited capability in producing new subjects. CemiFace often produces distorted face images, and those subjects exhibit a high similarity score.

Subjects exhibit the lowest diversity Fig. 20 shows the failure case images generated by conventional [2, 5, 29, 35, 40, 48, 55] and our methods. We report virtual subjects among those who exhibit the lowest 5% intra-class diversity in each dataset. Note that intra-class diversity does not pertain to changes in characteristics but relates to hard scenarios such as variations in pose, lighting conditions, occlusion, and resolution. As diversity evaluation employs FIQA methods, which assess the difficulty level of generated face images for recognition purposes, the resulting diversity score might not entirely align with human perception. Nevertheless, it can be observed that subjects with low diversity consist only of high-quality frontal faces. However, even low-diversity subjects generated with VIGFace have a high-quality yet diverse pose or lighting.

I. Identity leakage

To assess the potential for data or identity leakage in generative models, we search for the most similar face from the training dataset. For this experiment, we query the feature similarity for every sample in the training dataset. Fig. 11 shows that the most similar samples between the synthetic dataset and the corresponding training dataset. CemiFace generates facial images from identity embedding vectors, and it samples synthetic identity vectors from WebFace4M. Consequently, this results in CemiFace producing individuals with the actual person. Vec2Face fails to create nonexistent identity, resulting in HSFace having individuals that are nearly identical to those in WebFace4M. As shown in the figure, our virtual identities are free from identity leakage.

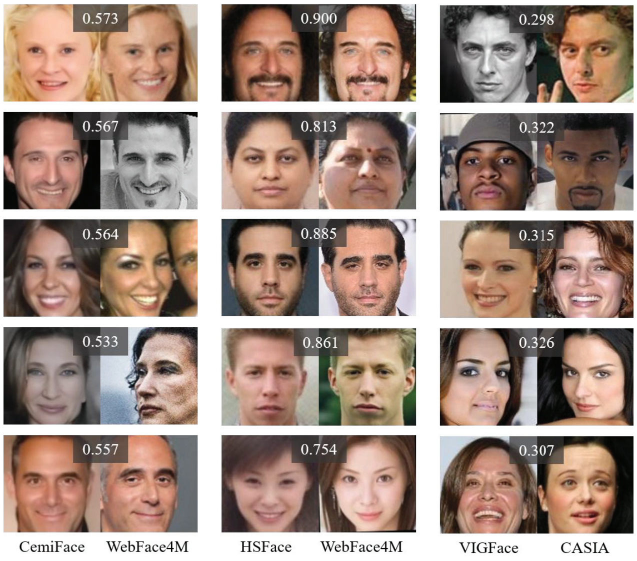


Figure 11. Training data leakage example. To assess the potential identity leakage, we search for the most similar face from the training dataset. We indicate the cosine similarity between the images.

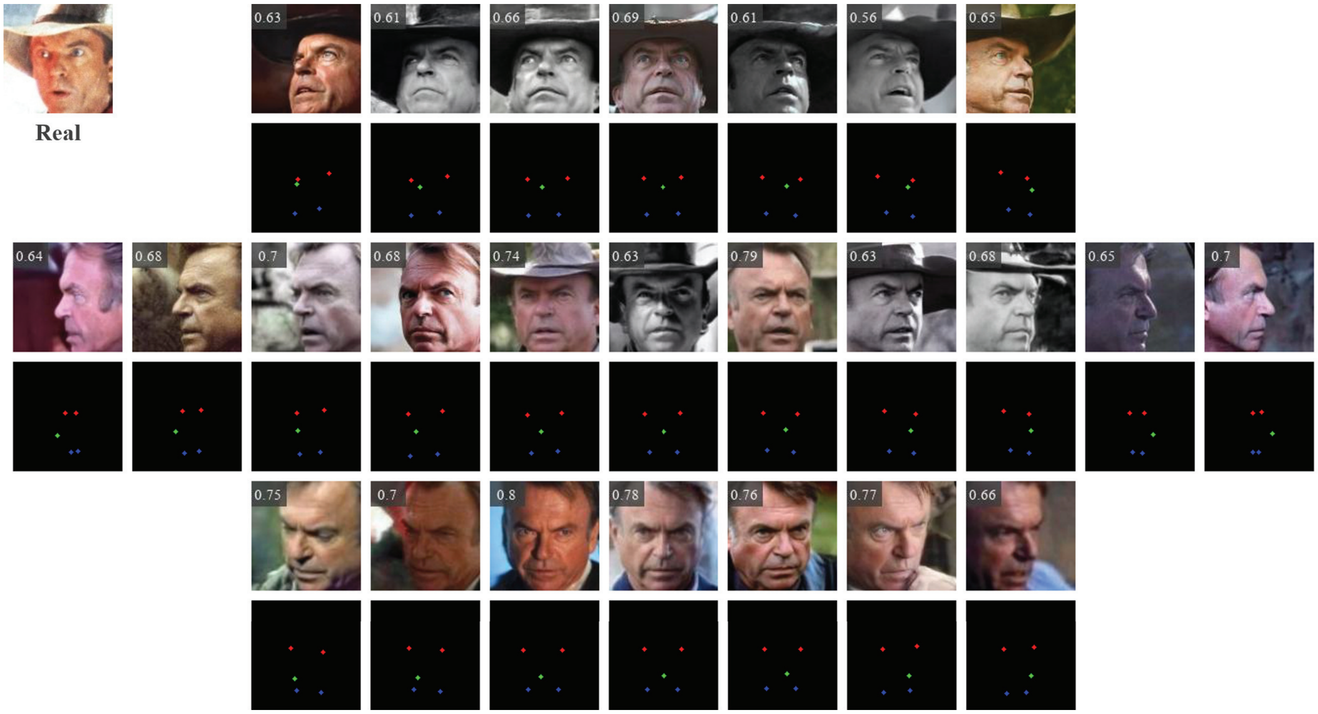


Figure 12. Multiview facial images from real ID

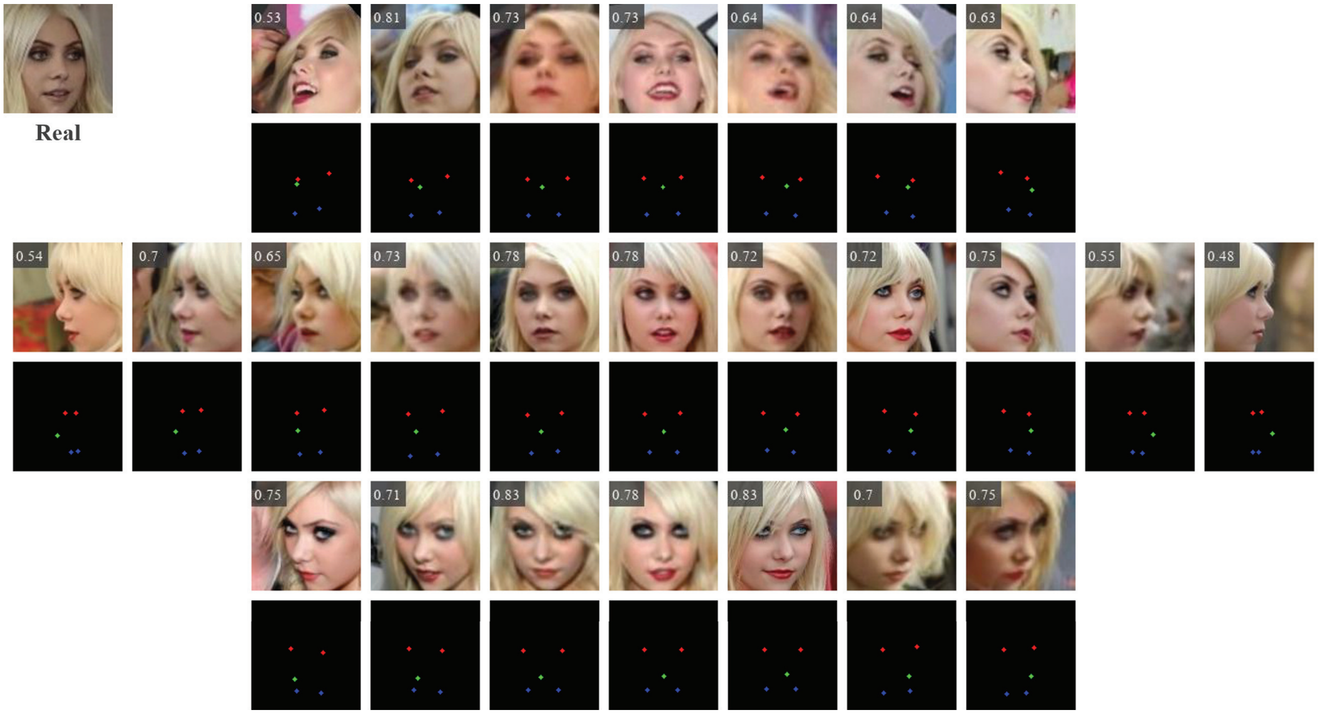


Figure 13. Multiview facial images from real ID

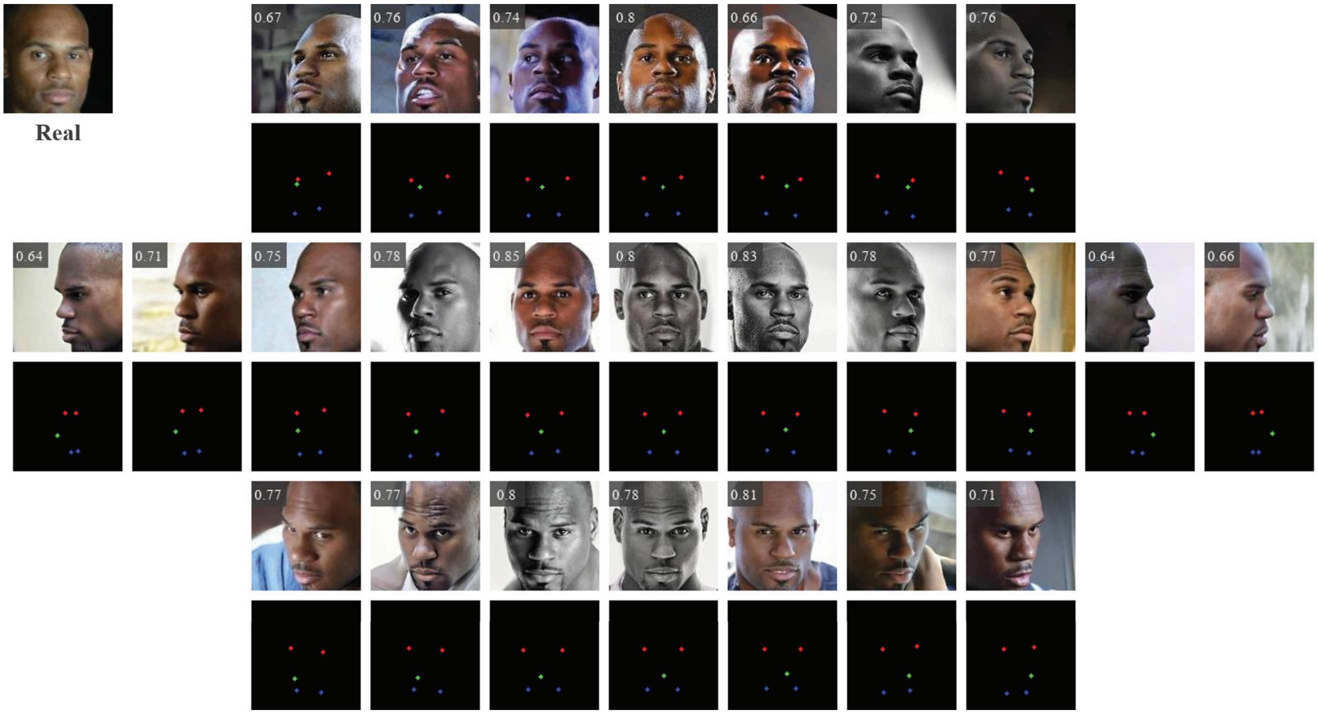


Figure 14. Multiview facial images from real ID

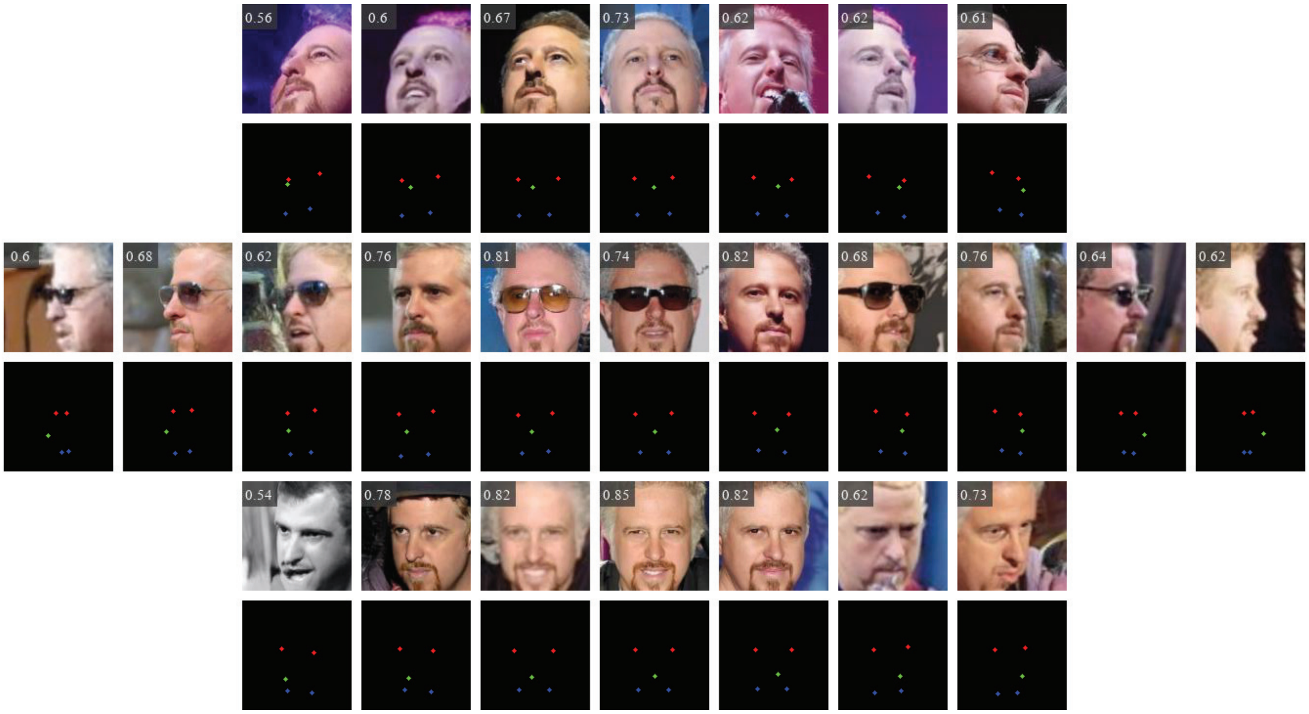


Figure 15. Multiview facial images from virtual ID

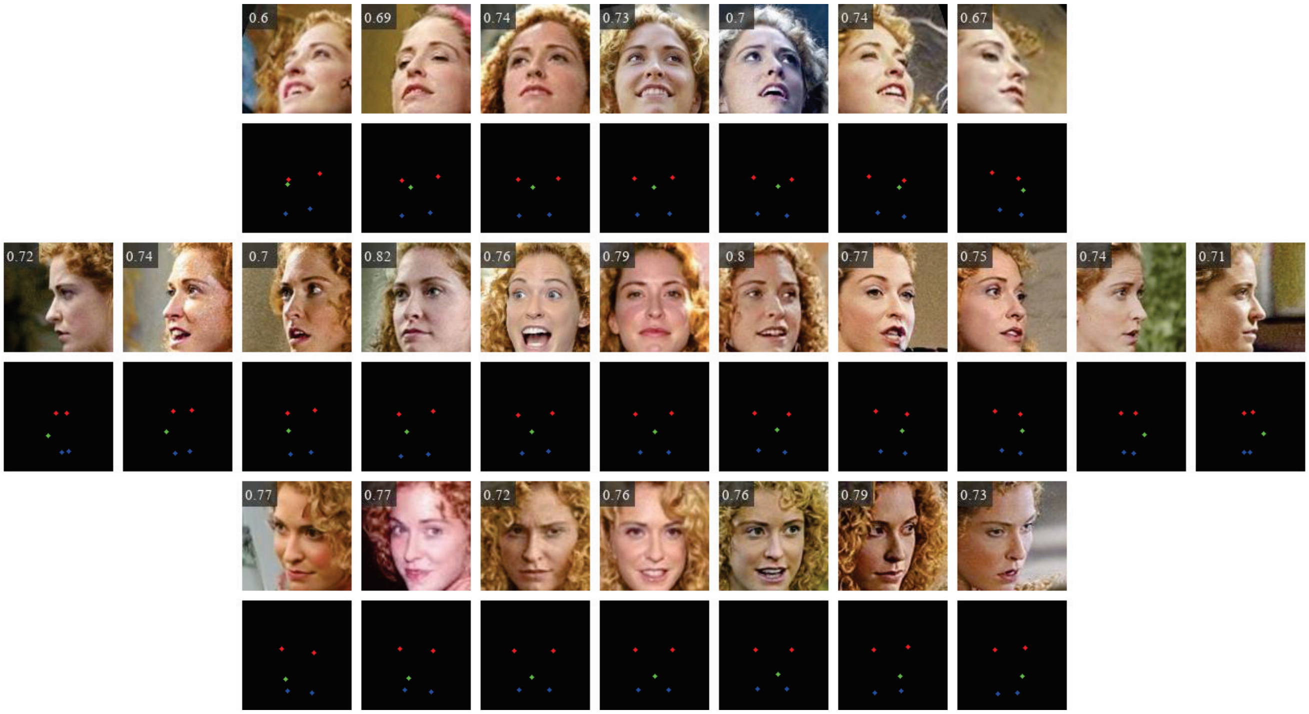


Figure 16. Multiview facial images from virtual ID

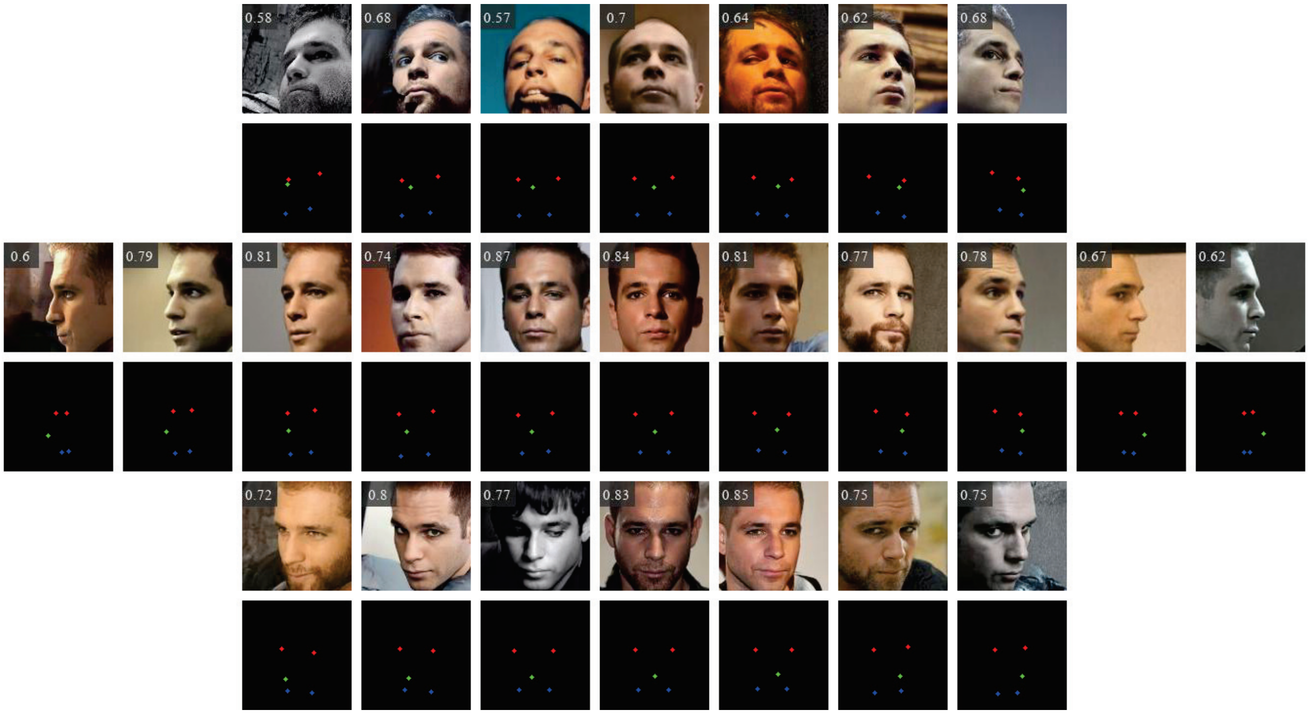


Figure 17. Multiview facial images from virtual ID

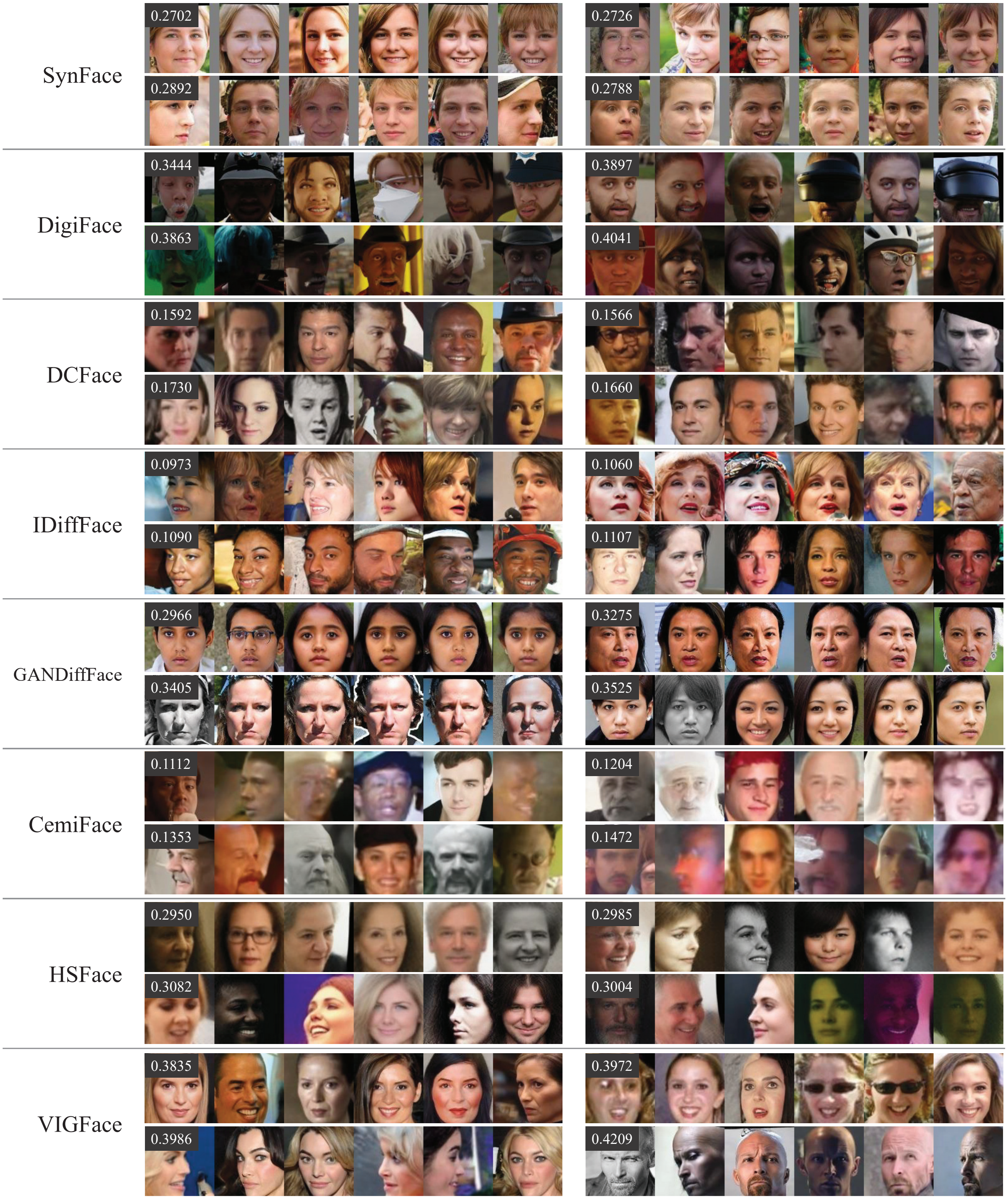


Figure 18. Low consistency subjects generated with various methods. Each row presents samples from the same ID that exhibits the lowest 5% consistency. The class consistency of each ID is indicated in the top-left corner.

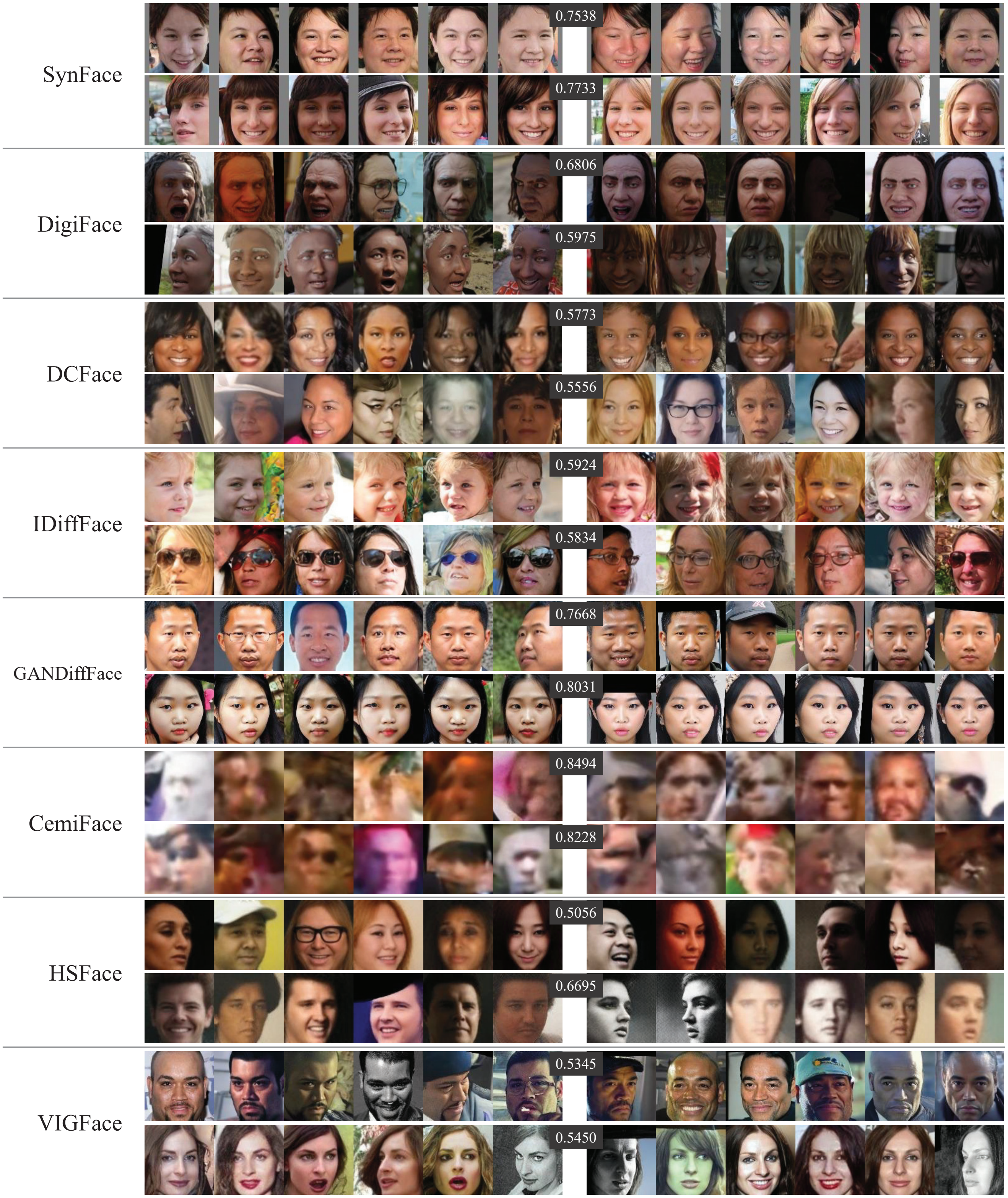


Figure 19. Inferior separability subjects generated with various methods. Each row presents samples from the top 5% most similar subject pairs, with the cosine similarity between the two subjects indicated.

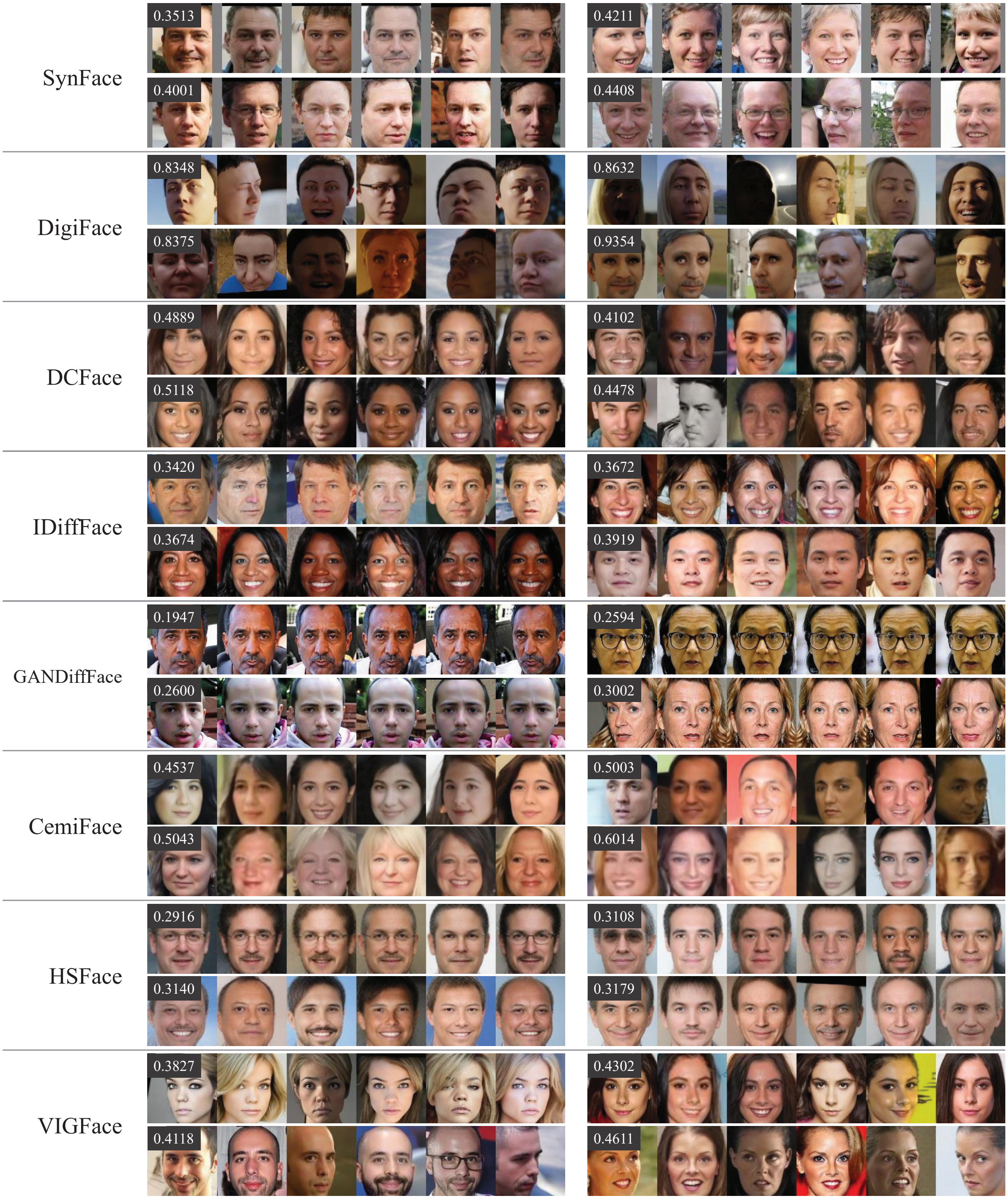


Figure 20. Low diversity subjects generated with various methods. Each row presents samples from the same ID that exhibits the lowest 5% diversity. The class diversity of each ID is indicated in the top-left corner.