Supplementary Material
DCFace: Synthetic Face Generation with Dual Condition Diffusion Model

Minchul Kim, Feng Liu, Anil K. Jain, Xiaoming Liu
Michigan State University, East Lansing, MI, 48824
{kimminc2, liufeng6, jain, liuxm}@cse.msu.edu

A. Training Details

A.1. Architecture Detals

The dual condition generator G_{mix} is a modification of DDPM [10] to incorporate two conditions. We insert two conditions $\boldsymbol{X}_{id}$ and $\boldsymbol{X}_{sty}$ into the denoising U-Net $\boldsymbol{\epsilon}_\theta(\boldsymbol{X}_t, t, \boldsymbol{X}_{id}, \boldsymbol{X}_{sty})$. Conditioning images $\boldsymbol{X}_{sty}$ and $\boldsymbol{X}_{id}$ are mapped to features using E_{sty} and E_{id}, respectively. According to Eq. 6 of the main paper, the style information $E_{sty}(\boldsymbol{X}_{sty})$ is the concatenation of style vectors at different $k \times k$ patch locations,

$$E_{sty}(\boldsymbol{X}_{sty}) := \boldsymbol{s} = \left[\boldsymbol{s}^1, \boldsymbol{s}^2, \boldsymbol{s}^{k_i} ..., \boldsymbol{s}^{k \times k}, \boldsymbol{s}'\right] \in \mathbb{R}^{(k^2+1) \times C}. \tag{1}$$

On the other hand, ID information is a concatenation of features extracted from a trainable CNN (e.g. ResNet50 [9]), which produces an intermediate feature $\boldsymbol{I}_{id}$ of shape $\mathbb{R}^{7 \times 7 \times 512}$and a feature vector $\boldsymbol{f}_{id}$ of shape $\mathbb{R}^{512}$. Specifically,

$$E_{id}(\boldsymbol{X}_{id}) := \boldsymbol{i} = [\text{Flatten}(\boldsymbol{I}_{id}), \boldsymbol{f}_{id}] + \boldsymbol{P}_{emb} \in \mathbb{R}^{50 \times C}, \tag{2}$$

where Flatten refers to removing the $H \times W$ spatial dimension and $\mathbb{R}^{50 \times C}$ is from concatenating features of length $7*7$ and 1. $\boldsymbol{P}_{emb}$ is a learnable position embedding for distinguishing each feature position for the subsequent cross-attention operation. Detailed illustrations of $E_{sty}(\boldsymbol{X}_{sty})$ and $E_{id}(\boldsymbol{X}_{id})$ are shown in Fig. 1. C for the channel dimension of $E_{sty}(\boldsymbol{X}_{sty})$ and $E_{id}(\boldsymbol{X}_{id})$ is 512.

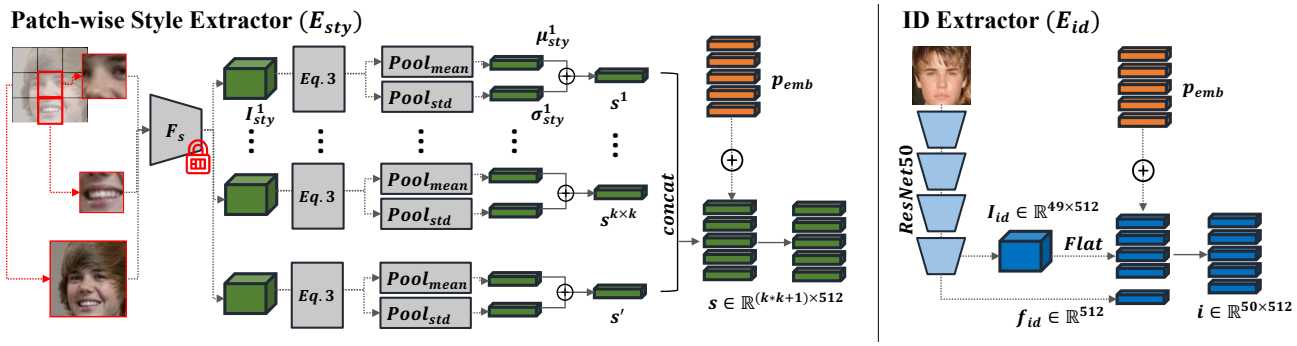

Figure 1. Left: An illustration of $\boldsymbol{X}_{sty}$. The key property of $\boldsymbol{X}_{sty}$ is in restricting the information in $\boldsymbol{X}_{sty}$ from flowing freely to the next layer. The fixed feature encoder $\boldsymbol{F}_s$ and the patch-wise spatial mean-variance operation destroy the detailed ID information while preserving the style of an image. We create an output of size $\mathbb{R}^{(k^2+1)\times C}$. Right: A simple CNN based on ResNet50. We take intermediate representation and the last feature vector and concatenate them together to create a output of size $\mathbb{R}^{50\times C}$.

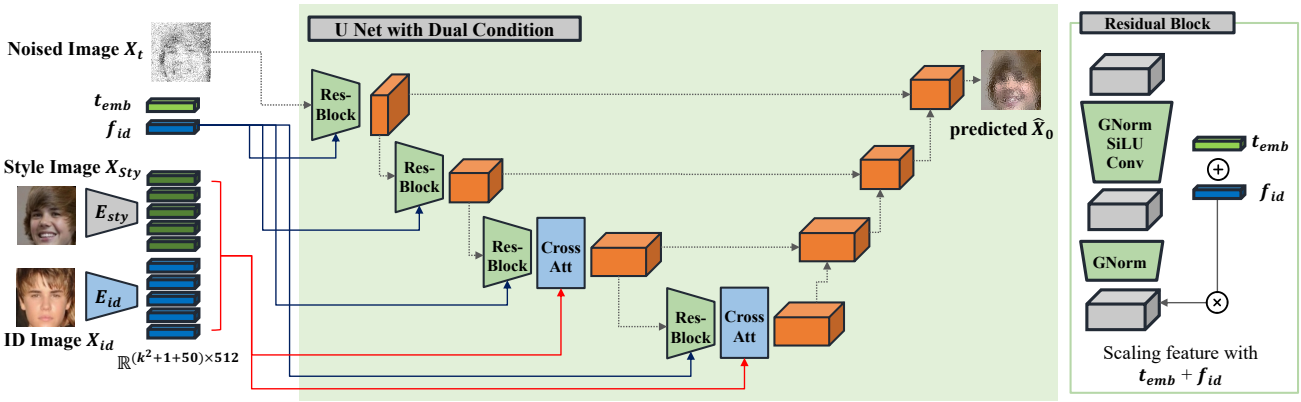

Figure 2. Illustration of DDPM U-Net with conditioning operations highlighted. The red arrow indicates how the dual conditions are injected into the intermediate features of U-Net using cross-attention layers. For clarity, up-sampling stages are not illustrated, but they are symmetric to the down-sampling stages. On the right is a detailed illustration of the Residual Block with timestep and ID condition. $\boldsymbol{t}_{emb}$ and $\boldsymbol{f}_{id}$ from E_{id} are added together and used to scale the output of the Residual Block.

When $E_{sty}(\boldsymbol{X}_{sty})$ and $E_{id}(\boldsymbol{X}_{id})$ is prepared, they together form $(k^2+1)+50$ vectors of shape 512. These can be injected into the U-Net $\boldsymbol{\epsilon}_\theta$ by following the convention of the DDPM based text-conditional image generators [18]. Specifically, cross attention operation can be written as a modification of attention equation [22] with query $\boldsymbol{Q}$, key $\boldsymbol{K}$ and value $\boldsymbol{V}$ with additional query $\boldsymbol{Q}_c$, key $\boldsymbol{K}_c$.

$$\text{Attn}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{SoftMax}\left(\frac{\boldsymbol{Q}\boldsymbol{W}_q(\boldsymbol{K}\boldsymbol{W}_k)^\intercal}{\sqrt{d}}\right)\boldsymbol{W}_v\boldsymbol{V}, \tag{3}$$

$$\text{Cross-Attn}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V},\boldsymbol{K}_c,\boldsymbol{V}_c) = \text{SoftMax}\left(\frac{\boldsymbol{Q}\boldsymbol{W}_q([\boldsymbol{K},\boldsymbol{K}_c]\boldsymbol{W}_k)^\intercal}{\sqrt{d}}\right)\boldsymbol{W}_v[\boldsymbol{V},\boldsymbol{V}_c], \tag{4}$$

where $\boldsymbol{W}_q, \boldsymbol{W}_k$ and $\boldsymbol{W}_v$ are learnable weights and $[\cdot]$ refers to concatenation operation. In our case, $\boldsymbol{Q} = \boldsymbol{K} = \boldsymbol{V}$ are an arbitrary intermediate feature in the U-Net. And $\boldsymbol{K}_c = \boldsymbol{V}_c$ are conditions generated by $E_{sty}(\boldsymbol{X}_{sty})$ and $E_{id}(\boldsymbol{X}_{id})$, concatenated together. This operation allows the model to update the intermediate features with the conditions if necessary. We insert the cross-attention module in the last two DownSampling Residual Blocks in the U-Net, as shown in Fig. 2.

To increase the effect of $\boldsymbol{X}_{id}$ in the conditioning operation, we also add $\boldsymbol{f}_{id}$ to the time-step embedding $\boldsymbol{t}_{emb}$. As shown in the right side of Fig. 2, the Residual Block in the U-Net modulates the intermediate features according to the scaling vector provided by $\boldsymbol{f}_{id} + \boldsymbol{t}_{emb}$. GNorm [25] refers to Group Normalization and SiLU refers to Sigmoid Linear Units [7]. Adding $\boldsymbol{f}_{id}$ to $\boldsymbol{t}_{emb}$ for the Residual Block allows more paths for $\boldsymbol{X}_{id}$ to change the output of U-Net.

A.2. Training Hyper-Parameters

The final loss for training the model end-to-end is $L_{MSE} + \lambda L_{ID}$ with λ as a scaling parameter. We set $\lambda = 0.05$ to compensate for the different scale between L2 and Cosine Similarity. All our input image sizes are 112×112, following the convention of SoTA face recognition model datasets [5, 12, 30]. And our code is implemented in Pytorch.

B. More Experiment Results

B.1. Adding Real Dataset

We include additional experiment results that involve adding real images. Although the motivation of the paper is to use an only-synthetic dataset to train a face recognition model, the performance comparison with an addition of a subset of the real dataset has its merits; it shows 1) whether the synthetic dataset is complementary to the real dataset and 2) whether the synthetic dataset can work as an augmentation for real images.

Tab. 1 shows the performance comparison between DigiFace [3] and our proposed DCFace when 1) a few real images are added and 2) both synthetic datasets are combined. The performance gap for DigiFace is large, jumping from 86.37 to 92.67 on average when $2K$ real subjects with 20 images per subject are added. In contrast, ours show a relatively less dramatic gain, 91.21 to 92.90 when few real images are added. This indicates that DigiFace [3] is quite different from the real images and ours is similar to the real images. This is in-line with our expectation as we have created a synthetic dataset that tries to mimic the style distribution of the training dataset, whereas DigiFace simulates image styles using 3D models.

B.2. Combining Multiple Synthetic Datasets

In the second to the last row of Tab. 1, when we combined the two synthetic datasets without the real images, the performance is the highest, reaching 93.06 on average. This result indicates that different synthetic datasets can be complementary when they are generated using different methods.

	# Synthetic Imgs	# Real Imgs	LFW	CFPFP	CPLFW	AGEDB	CALFW	AVG	Gap to Real
DigiFace	1.2M (10K×72+100K×5)	0	96.17	89.81	82.23	81.10	82.55	86.37	8.72
DigiFace	1.2M (10K×72+100K×5)	2K×20	99.17	94.63	88.1	90.5	90.97	92.67	2.06
DCFace	1.2M (20K×50+40K×5)	0	98.58	88.61	85.07	90.97	92.82	91.21	3.61
DCFace	1.2M (20K×50+40K×5)	2K×20	98.97	94.01	86.78	91.80	92.95	92.90	1.82
DCFace+DigiFace (2.4M)		0	99.20	93.63	87.25	92.25	92.95	93.06	**1.65**
CASIA	0	0.5M	99.42	96.56	89.73	94.08	93.32	94.62	0

Table 1. Verification accuracies of FR models trained with synthetic datasets and subset of real datasets. In all settings, the backbone is set to IR50 [5] model with AdaFace loss [16] for a fair comparison.

C. Analysis

C.1 Unique Subject Counts. In Fig. 3, we plot the number of unique subjects that can be sampled as we increase the sample size. The blue curve shows that the number of unique samples that can be generated by a DDPM of our choice does not saturate when we sample $200,000$ samples. At $200,000$ samples, the unique subjects are about $60,000$. And by extrapolating the curve, we estimate the number might reach $80,000$ with more samples. Our DDPM of choice is trained on FFHQ [15] dataset which contains $70,000$ unlabeled high-quality images. The orange line shows the number of unique samples that are sufficiently different from the subjects in the CASIA-WebFace dataset. The green line shows the number of unique samples left after filtering images that contain sunglasses. The flat region is due to the filtering stage reducing the total candidates. The plot shows that DDPM trained on FFHQ dataset can sufficiently generate a large number of unique and new samples that are different from CASIA-WebFace dataset. However, with more samples, eventually there is a limit to the number of unique samples that can be generated. When the number of total generated samples is $100,000$, one additional sample has approximately 24% chance of being unique, whereas, at $200,000$, the probability is 15%. The rate of sampling another unique subject decreases with more samples. The model used for evaluating the uniqueness is IR101 [5] trained on the WebFace4M [30] dataset. And we use the threshold of 0.3. We would like to note a typo in Sec. 3.3 of the main paper, where the number of unique subjects should be corrected from $62,570$ to $42,763$.

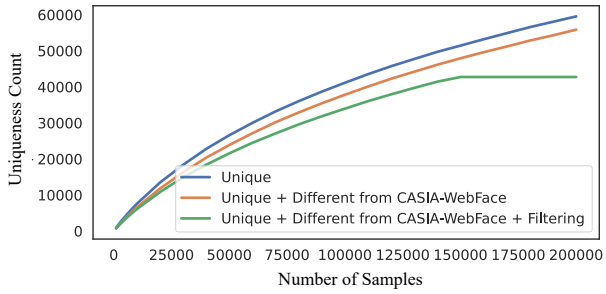

Figure 3. Plot of unique subject count as the number of samples from G_{id} is increased from 1000 to $200,000$. At $200,000$, one additional sample has approximately 15% chance of being unique. And the rate decreases with more samples.

C.2 Feature Plot. In Fig. 4, we show the 2D t-SNE [21] plot of synthetic images generated by 3 different methods (DiscoFaceGAN [6], DigiFace [3] and proposed DCFace). The red circles represent real images from CASIA-WebFace. We extract the features from each image using a pre-trained face recognition model, IR101 [5] trained on WebFace4M [30]. We show two settings we sample (a) 50 subjects with 1 image per subject and (b) 1 subject with 50 images per subject. Note that the proximity of DCFace image features is closer to CASIA-WebFace image features, highlighted in a circle. For each setting, we show the features extracted from an intermediate layer of IR101 and the last layer. As the layer becomes deeper, the features become suitable for recognition, as shown in the last column of the figure.

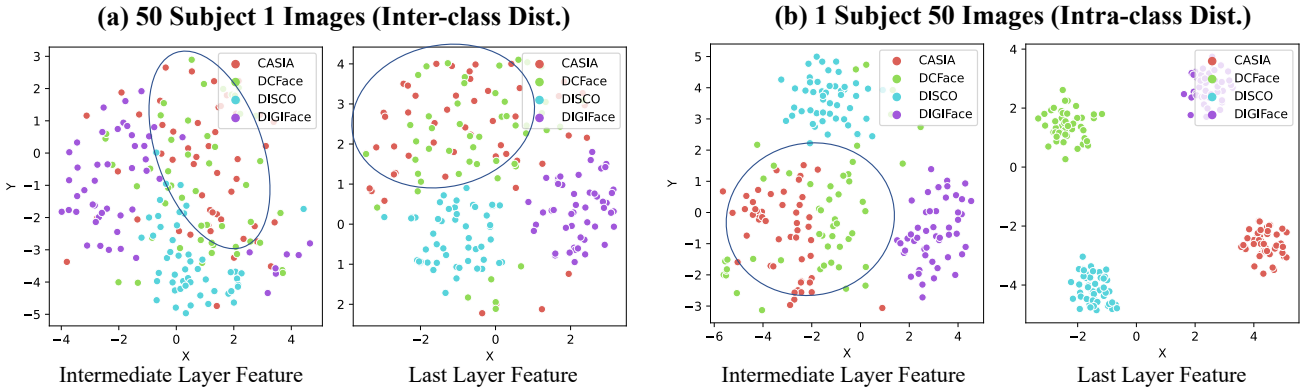

Figure 4. (a) the t-SNE plot of features from synthetic and real datasets of 50 subjects per dataset. It shows how 50 randomly sampled subjects from each dataset are distributed. The distribution between real (red) and DCFace (green) is the closest. (b) the t-SNE plot of features from synthetic and real datasets of 1 subject per dataset with 50 images. We randomly sample 1 subject from each dataset. The last layer features are well separated as the model is a face recognition model that separates the features of different subjects.

C.3 Comparison with Classifier Free Guidance.

When $\epsilon(x_t, c)$ learns to use the condition c, the difference $\epsilon(x_t, c) - \epsilon(x_t)$ can give further guidance during sampling to increase the dependence on c. But, in our case, the ID condition is the fine-grained facial difference that is hard to learn with MSE loss. Proposed Time-dependent ID loss, L_{ID} helps the model learn this directly. Row 3 vs 4 of Tab. 2 shows that L_{ID} is more effective than CFG.

	Conditions	Train Loss	Sampling	FR.Perf ↑
1	CNN(X_{id}), CNN(X_{sty})	MSE	+ Guide	73.38
2	CNN(X_{id}), E_{sty}(X_{sty})	MSE	×	82.30
3	CNN(X_{id}), E_{sty}(X_{sty})	MSE	+ Guide	84.05
4	CNN(X_{id}), E_{sty}(X_{sty})	MSE+L_{ID}	×	**89.56**

Table 2. Green E_{sty} and L_{ID} indicates the novelty of our paper. For guidance, we adopt 10% condition masking during training and the guidance scale of 3 during sampling. FR.Perf is an average of 5 face recognition performances as in the main paper.

Interestingly, with a large guidance scale, CFG becomes harmful. CFG decreases diversity as pointed out by [11]. We observe that guidance with X_{id} leads to consistent ID but with little facial variation, the same phenomenon in DCFace with grid-size 1x1 in E_{sty}, in Tab. 2 (main). Good FR datasets need both large intra and inter-subject variability and we combine E_{sty} and L_{ID} to achieve this.

C.4 FID Scores. Note that our generated data is not high-res images like FFHQ when compared to how SynFace is similar to FFHQ. (Tab. 3 row 5 vs 6). But, we point out that our aim is not to create HQ images but to create a *database* with realistic inter/intra-subject variations. In that regard, we have successfully approximated the distribution of the popular FR training dataset CASIA-WebFace (FID=13.67).

	Generator Train Data	Source (real/syn)	Target (real)	FID ↓
1	-	CASIA (train)	CASIA (val)	**9.57**
2	CASIA (train)	DCFace	CASIA (val)	**13.67**
3	FFHQ+3DMM	SynFace	CASIA (val)	38.48
4	3D Face Capture	DIGIFACE1M	CASIA (val)	71.65
5	CASIA (train)	DCFace	FFHQ (train+val)	35.45
6	FFHQ+3DMM	SynFace	FFHQ (train+val)	**21.75**
7	3D Face Capture	DIGIFACE1M	FFHQ (train+val)	68.67

Table 3. FID scores of synthetic vs real datasets. For synthetic datasets, we randomly sampled $10,000$ images. See Line 630 for Casia-WebFace Train and Val set split. All images are aligend and cropped to 112×112 to be in accordance with CASIA-WebFace.

Having said this, we note FID is not comprehensive in evaluating labeled datasets. It cannot capture the label consistency nor directly relate to the FR performance. As such, SynFace/DigiFace do not report FID. We propose U,D,C metrics that enable holistic analysis of labeled datasets.

C.5 Does DCFace change gender?. DCFace combines X_{ID} and X_{sty}, while adhering to the subject ID as defined by a pre-trained FR model. Factors weakly related to ID, such as age and hair style, can vary. Biometric ambiguity can occur due to makeup, wig, weight change, *etc.* even in real life. The perceived gender may change, but changes such as hair are less relevant to subject ID for the FR model.

C.6 Why DCFace is better in U,D,C metrics?. We note DCFace is not better in all U,D,C. Fig. 6 (main) shows SynFace has the highest consistency (C). But, DCFace excels in the tradeoff between C and D. In other words, style similarity to the real dataset (*i.e.* D) is lacking in other datasets and it is as important as ID consistency. As such, U,D,C metrics reveal weak/strong points of synthetic datasets.

D. Visualizations

D.1. Time-step Visualizaton

Fig. 5 shows how DDPM generates output at each time-step. The far left column shows $\boldsymbol{X}_{sty}$, the desired style of an image. The far right column shows $\boldsymbol{X}_{id}$, the desired ID image of choice. In early time-steps, the network reconstructs the front-view image with an ID of $\boldsymbol{X}_{id}$. And gradually, it interpolates the image into the desired style of $\boldsymbol{X}_{sty}$. The gradual transition can be in the pose, hair-style, expression, etc.

Figure 5. A plot of DCFace outputs at each time-step.

D.2. Interpolation

In Fig. 6, we show the plot of interpolation in $\boldsymbol{X}_{sty}$. While keeping the same identity $\boldsymbol{X}_{id}$, we take two style images $\boldsymbol{X}_{sty1}$ and $\boldsymbol{X}_{sty2}$. We interpolate with α in $\alpha E_{stry}(\boldsymbol{X}_{sty1}) + (1-\alpha)E_{stry}(\boldsymbol{X}_{sty2})$ with α increasing linearly from 0 to 1. The interpolation is smooth, creating an intermediate pose and expression that did not exist before.

Figure 6. A plot of DCFace output with style interpolation.

E. Miscelaneous

Similarity threshold. Threshold=0.3 is based on FR evaluation model having a threshold of 0.3080 for verification with TPR@FPR=0.01% : 97.17% on IJB-B [24]. FPR=0.01% is widely used in practice and the scale of similarity is $(-1, 1)$. At threshold=0.3, FFHQ has 200 (2%) more unique subjects than DDPM, signaling a similar level of uniqueness.

Style Extracting Model. We use the early layers of face recognition model for style extractor backbone. Our rationale for adopting the early layers of the FR model, as opposed to that of the ImageNet-trained model is that the early layers extract low-level features and we wanted features optimized with the face dataset. But, it is possible to take other models as long as it generates low-level features.

Evaluation on Harder Datasets. We evaluate on harder datasets, IJB-B [24] (TPR@FPR=0.01%: 75.12) and TinyFace [4] (Rank1: 41.66). We include this result for future works to evaluate on harder datasets.

Real and Generated Similarity Analysis. In addition to Fig.7 mathcing $\hat{X}_{id}$ with CASIA-WebFace, matching all $\hat{X}_0$ (generated) images against CASIA-WebFace at threshold=0.3, we get 0.0026% FMR. This implies that only a small fraction of CAISA-WebFace images are similar to the generated images.

F. Societal Concerns

We believe that the Machine Learning and Computer Vision community should strive together to minimize the negative societal impact. Our work falls into the category of 1) image generation using generative models and 2) synthetic labeled dataset generation. In the field of image generation, unfortunately, there are numerous well-known malicious applications of generative models. Fake images can be used to impersonate high-profile figures and create fake news. Conditional image generation models make the malicious use cases easier to adapt to different use cases because of user controllability. Fortunately, GAN-based generators produce subtle artifacts in the generated samples that allow the visual forgery detection [2, 8, 23, 26]. With the recent advance in DDPM, the community is optimistic about detecting forgeries in diffusion models [20]. It is also known that proactive treatments on generated images increase the forgery detection performance [2], and as generative models become more sophisticated, proactive measures may be advised whenever possible.

Synthetic dataset generation is, on the other hand, an effort to avoid infringing the privacy of individuals on the web. Large-scale face dataset is collected without informed consent and only a few evaluation datasets such as IJB-S [14] has IRB compliance for safe and ethical research. Collecting large-scale datasets with informed consent is prohibitively challenging and the community uses web-crawled datasets for the lack of an alternative option. Therefore, efforts to create synthetic datasets with synthetic subjects can be a practical solution to this problem. In our method, we still use real images to train the generative models. We hope that research in synthetic dataset generation will eventually replace real images, not just in the recognition task, but also in the generative tasks as well, removing the need for using real datasets in any form.

G. Implementation Details and Code

The code will be released at https://github.com/mk-minchul/dcface. For preprocessing the training data CASIA-WebFace [12], we reference AdaFace [16] and use MTCNN [27] for alignment and cropping faces. For the backbone model definition, TFace [1] and for evaluation of LFW [13], CFP-FP [19], CPLFW [28], AgeDB [17] and CALFW [29], we use AdaFace repository [16].

References

[1] TFace. https://github.com/Tencent/TFace.git. Accessed: 2021-10-3. 7

[2] Vishal Asnani, Xi Yin, Tal Hassner, Sijia Liu, and Xiaoming Liu. Proactive image manipulation detection. In *CVPR*, 2022. 7

[3] Gwangbin Bae, Martin de La Gorce, Tadas Baltrusaitis, Charlie Hewitt, Dong Chen, Julien Valentin, Roberto Cipolla, and Jingjing Shen. Digiface-1m: 1 million digital face images for face recognition. In *WACV*, 2023. 3, 4

[4] Zhiyi Cheng, Xiatian Zhu, and Shaogang Gong. Low-resolution face recognition. In *ACCV*, 2018. 7

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *CVPR*, 2019. 2, 3, 4

[6] Yu Deng, Jiaolong Yang, Dong Chen, Fang Wen, and Xin Tong. Disentangled and controllable face image generation via 3D imitative-contrastive learning. In *CVPR*, 2020. 4

[7] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107, 2018. 2

[8] Sharath Girish, Saksham Suri, Sai Saketh Rambhatla, and Abhinav Shrivastava. Towards discovery and attribution of open-world gan generated images. In *ICCV*, 2021. 7

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33, 2020. 1

[11] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 5

[12] Gary Huang, Marwan Mattar, Honglak Lee, and Erik Learned-Miller. Learning to align from scratch. *NeurIPS*, 25, 2012. 2, 7

[13] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled Faces in the Wild: A database forstudying face recognition in unconstrained environments. In *Workshop on Faces in'Real-Life'Images: Detection, Alignment, and Recognition*, 2008. 7

[14] Nathan D Kalka, Brianna Maze, James A Duncan, Kevin O'Connor, Stephen Elliott, Kaleb Hebert, Julia Bryan, and Anil K Jain. IJB–S: IARPA Janus Surveillance Video Benchmark. In *BTAS*, 2018. 7

[15] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 4

[16] Minchul Kim, Anil K Jain, and Xiaoming Liu. AdaFace: Quality adaptive margin for face recognition. In *CVPR*, 2022. 3, 7

[17] Stylianos Moschoglou, Athanasios Papaioannou, Christos Sagonas, Jiankang Deng, Irene Kotsia, and Stefanos Zafeiriou. AGEDB: the first manually collected, in-the-wild age database. In *CVPRW*, 2017. 7

[18] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[19] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *WACV*, 2016. 7

[20] Zeyang Sha, Zheng Li, Ning Yu, and Yang Zhang. De-fake: Detection and attribution of fake images generated by text-to-image diffusion models. *arXiv preprint arXiv:2210.06998*, 2022. 7

[21] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 2008. 4

[22] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[23] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *CVPR*, 2020. 7

[24] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn Adams, Tim Miller, Nathan Kalka, Anil K Jain, James A Duncan, Kristen Allen, et al. IARPA Janus Benchmark-B face dataset. In *CVPRW*, 2017. 7

[25] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018. 2

[26] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *ICCV*, 2019. 7

[27] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *Signal Processing Letters*, 2016. 7

[28] Tianyue Zheng and Weihong Deng. Cross-Pose LFW: A database for studying cross-pose face recognition in unconstrained environments. *Beijing University of Posts and Telecommunications, Tech. Rep*, 5, 2018. 7

[29] Tianyue Zheng, Weihong Deng, and Jiani Hu. Cross-Age LFW: A database for studying cross-age face recognition in unconstrained environments. *CoRR*, abs/1708.08197, 2017. 7

[30] Zheng Zhu, Guan Huang, Jiankang Deng, Yun Ye, Junjie Huang, Xinze Chen, Jiagang Zhu, Tian Yang, Jiwen Lu, Dalong Du, et al. WebFace260M: A benchmark unveiling the power of million-scale deep face recognition. In *CVPR*, 2021. 2, 4