

Generating Frontal Face to Improve Side Face Recognition

Yuxiang Guo, Yuguang Sun, Deming(Remus) Li

May 6, 2022

Abstract

Unconstrained frontal face recognition has made great advances in recent years, benefiting from rapid development of deep learning models and an increasing variety of datasets. For example, [1] has achieved 99% accuracy on Extended Yale-B dataset with Deep Multimodal Subspace Clustering Networks. Namely, these models are strong and robust to frontal-view face recognition. However, there remain challenges and constraints that limit the models' performance in real-world settings. Pose variation is obviously one of them. In this project, we first verify this challenge through face recognition using Vision Transformer (ViT) [27] on two datasets - FEI[6] and CPLFW[26]. Then instead of directly addressing the recognition rate of side-view face images, we employ generative models to generate frontal view synthesis from a single face image in a self-supervised manner [28]. Through experiments, side-view face images are harder to recognized and identified, especially in extreme-pose and unconstrained setting. Although our final recognition results are not improved through our rotation model due to several limitations that we acknowledge in our report, our generative model is able to synthesize relatively realistic and reasonable frontal face images.

1. Introduction

The recent development of deep learning keeps improving the accuracy of image classifications for various data types. Deep learning models keeps the state of the art accuracy among several traditional data sets. In the face verification task of LFW [10], models such as SphereFace [17] and VGGFace[16] utilizing Convolutional Neural Network (CNN), have achieved a nearly perfect accuracy. Although those deep learning models are promising in these unconstrained environments or controlled settings, many of them failed to maintain a good performance for recognizing faces of different poses. For instance, the accuracy of SphereFace and VGGFace on CPLFW [26] dataset, which is much more pose-variant, drops about 20% compared that on LFW dataset. Intuitively speaking, it is also harder to recognize from face profile than front face. Trying to overcome such difficulty, several researchers propose more robust models for those side faces, but others raise the idea of generating frontal face from side face, and then recognize them. In this project, we manage to adapt and modify 3-D

face rotation [28] to generate frontal face, and throughout the paper, we use Face ViT[27], which is famous recently among NLP and other computer vision tasks, to serve as recognition model which we manage to improve its side face recognition accuracy. We do the test on a subset of CPLFW which contains most extreme case and achieving 5.4% recognition rate.

2. Related Work

2.1. Models for Face Recognition

Face recognition task has been discussed for a long time, and thanks to the development of the deep learning models, computer hardware and large-scale training databases such as ImageNet [4], nowadays we have deeper insight and better results in this area. Based on convolution neural networks, VGGFace, ResNet[9] are two batchmarks for face recognition. Deep CNN models, such as DeepFace[20] which uses 9-layer CNN, and combined ResNet such as SphereFace, which implements 64-layer ResNet, achieved a nearly perfect accuracy on unconstrained test dataset LFW.

Although it has been tested that CNN-based model can perform well on face recognition tasks, their robustness on different dataset drives the search for newer structure. Recently, Transformer models[22] have demonstrated great performance on a broad range of Natural Language Processing (NLP) tasks. Since visual data follows a typical structure which is similar to word embeddings, after careful network designs, training schemes and large-scale of training dataset, transformer and its variants have been successfully applied in vision area. Transformers are based on the multi-head self-attention module, which learns the relationships among input elements. The significant difference with Transformer and Recurrent Neural Network (RNN) models is that RNN-based models can only learn relationships of short-term context, but self-attention module can attend to the whole inputs. Based on this, ViT[7] splits the image into 16 pieces, and transfers them into appropriate vectors with a position token. Similar to BERT[5], ViT attaches the whole input with a learnable class token, which serves as the final prediction. After ViT, several variants are applied successfully for image segmentation, object detection, etc..

In this paper, we applied Face Transformer[27], which is the Transformer model modified for the face recognition task, to serve as the recognition model. The model can achieve a great result on LFW dataset, but as for the pose-

variant face, it cannot perform well. To improve the side face recognition performance, we apply the face frontalization step, i.e. generating frontal face based on side face image.

2.2. Models for Face Generating

2.2.1 Multi-View Synthesis

For the first time, DR-GAN [21] adopts GAN to generate frontal faces with encoder-decoder structure, but with perceptually visible artifacts. Then TP-GAN [11] utilizes global and local features with multi-task learning strategy to frontalize faces. These methods are trained on Multi-PIE dataset, which makes them overfit the controlled environment.

Several attempts have been made to incorporate 3D prior knowledge into GAN-based frontalization pipeline. FF-GAN [25] proposes to integrate the regression of 3DMM coefficients into the network and employ them for generation. UV-GAN [3] proposes to complete UV-map using image-to-image translation. However, these pipelines requires high precision 3D fitting and ground-truth Uv-maps. Recently, HF-PIM [2] achieves high-quality face frontalization results using facial texture map and correspondence fields. However their method also requires paired data for training.

2.2.2 Image-to-Image Translation

Image-to-image translation aims at translating an input image to a corresponding output image such as domain adaptation or translation. Pix2Pix framework [12] first used image-conditional GANs for this task. Pix2PixHD [23] used stacked structures to produce high-quality images. Recently, SPADE [18] and MaskGAN [15] discovers that semantic information can be infused via conditional batch normalization to further improve the results.

Cycle consistency has been proven useful on various tasks, especially in unpaired data settings. CycleGAN [29] achieves unpaired image-to-image translation, and GANimation [19] proposes to generation animation without supervision.

3. Approach

3.1. Vision Transformer for Face Recognition

In model design, Face Transformer follows the original transformer to preserve its advantages on simple setup, computational efficiency and scalable. The model contains three subparts: image embedding, transformer encoder and decoder and prediction.

3.1.1 Image Embedding

Since the original input of NLP transformer is word embedding, a 1-d vectors, we first need to change the 3-d(2-d with channels) images into appropriate 1-d vectors. We reshape the image $x \in R^{H \times W \times C}$ into a sequence of flattened 2-d patches. We split them into 16 patches with overlapping to preserve as much information as possible. Then, after flatten the patches, we map them to 1-d dimensions with a trainable linear projection.

Similar to NLP transformer, we prepend a position token with each patch to stand for their particular position in the whole picture. Inspired by BERT's [class] token, we also append a learnable embedding to the sequence of embedded patches, whose state at the output of the transformer encoder serves as the image representation y . We use this token as the final prediction of the input patches and apply it to loss function to implement forward-backward update method.

3.1.2 Transformer Encoder and Decoder

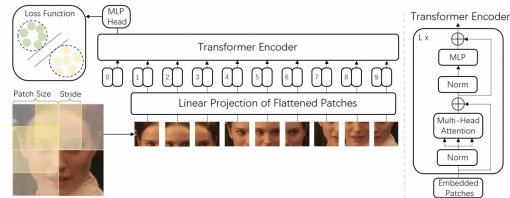


Figure 1. Overall Structure of Face Transformer

For the encoder part, as shown in Figure 1, we use 6 identical encoders stacked together, each one is composed by one self-attention layer and a feed forward layer with two subsequent normalize layer to deal with the input patches. Also, the residual is used for each normalize layer.

We implement multi-head attention for each self-attention module. Multi-head attention is based on the scaled dot-product attention, which is calculated by $Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V$, where Q is the Query vector, K is the Key vector and V is the Value vector. Scaling factor $\frac{1}{\sqrt{d_k}}$ is applied here in order to avoid pushing the softmax function into regions where it has extremely small gradients.

Multi-head attention is composed by $h = 8$ parallel attention layers, which is calculated by

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O$$

where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$.

(1)

By this multi-head techniques, the model's ability to focus on different locations is expended, and multiple 'rep-

resentation subspaces' for the attention layer are provided, with no additional computational cost.

The decoder is also composed by a stack of $N = 6$ identical layers, for each layer it is composed by three sub-layers, specifically, one additional multi-head attention layer is added. Similar to the encoder, we implement residual connections around each sub-layers, and for the additional attention layer, we also modify it to prevent positions from attending to subsequent positions. This can ensure that the predictions for position i can depend only on the known outputs at positions less than i .

3.1.3 Prediction

After the multi-head attention modules, the fully connected layer and softmax layer are applied to modify the attention modules' output. The final output is supervised by an elaborate loss function for better discriminate ability,

$$L = -\log P_y = -\log \frac{e^{W_y^T x + b_y}}{\sum_{j=1}^C e^{W_j^T x + b_j}}, \quad (2)$$

where y is the label, P_y is the prediction of assigning x to class y , C is the number of identities, W_j is the j -th column of the weight of the last layer and b_j is the bias.

3.2. Frontal Face Generation

The current generative models heavily rely on datasets with multi-view images of the same person, which limits the scope of data source and applications. In the paper, the authors [28] proposes a novel unsupervised learning framework that can use only single-view images to synthesize photo-realistic rotated faces. The key idea is that **rotate** the 3D face back and forth, and **re-rendering** them to the 2D plane, which can be served as a strong self-supervision. The whole frame work consists of three parts: 3D face fitting, the rotate-and-render strategy, and the render-to-image generation module.

3.2.1 3D Face Fitting and Rendering

The method relies on rough 3D parametric fitting of given face, where all kinds of 3D models are applicable. Given 3D face model with n vertices, $V = [v_1, v_2, \dots, v_n]$ represent the normalized positions in 3D space. The projection of a 3D shape onto the 2D image can be written as:

$$\Pi(V, P) = f * p_r * R * V + h_{2d},$$

in which "*" is matrix multiplication. p_r is the orthographic projection matrix. R is the rotation matrix. And h_{2d} is the 2D offset.

For each vertex v_i , there exists an associated texture $t_i = [r_i, g_i, b_i]^T$. The simplest vertical projection to get the

colors of the vertices from the original image I . The color of each vertex can be represented as $t_i = I(\Pi(v_i, P))$.

Given a set of 3D representation of a face $\{V, P, T\}$, rendering is to map it to the 2D space and generate an image. Since there might exist multiple rotated matrix on the line $x = x_j$ and $y = y_j$, then the same texture will be assigned. For all vertice on the line, only the outermost one with the largest z axis value gets the correct texture.

Open-sourced Neural Mesh Renderer [14] is used to perform rendering without any training.

3.2.2 Rotate-and-render Training Strategy

Training pairs called Rotate-and-Render (R&R) which consists of two rotate-and-render operations. The key idea is to create the artifacts caused by rotating occluded facial surface to the front and eliminate them in a self-supervised manner.

Given an input image I_a , we first get its 3D model parameters P_a by a 3D-face fitting model. With a being the current view in the 2D space, the textures of vertices can be acquired as:

$$T_a = \text{GetTex}(I_a, \{V, P_a\}).$$

We then rotate the 3D representation of this face to another random 2D view b by multiplying R_a with R_{random} to get P_b . And we render the current 3D representation to $Rd_b = \text{Render}(\{V, P_b, T_a\})$, which is the first complete rotate-and-render operation.

$$T_b = \text{GetTex}(I_b, \{V, P_b\}).$$

Unlike previous approaches relying on ground truth image I_b , the authors propose recover T_a regarding T_b as input. Specifically, 3D position P_b is rotated back to P_a and render it back to its original 2D positions with:

$$Rd_{a'} = \text{Render}(\{V, P_a, T_b\}).$$

This $Rd_{a'}$ is basically a rendered image with artifacts caused by rotating a face from view b to a in the 2D space, which compared with the original image I_a

3.2.3 Render-to-Image Generation

In order to eliminate the artifacts and map the rendered images Rd_b and $Rd_{a'}$ from the rendered domain to real image domain, render-to-image generation module to create $Fa' = G(Rd_{a'})$ and $Fb = G(Rd_b)$ using generator G , as shown in Fig 2.

The generator G is adopted from CycleGAN [29]. The multi-layer discriminator and perceptual loss are borrowed from Pix2PixHD [23].

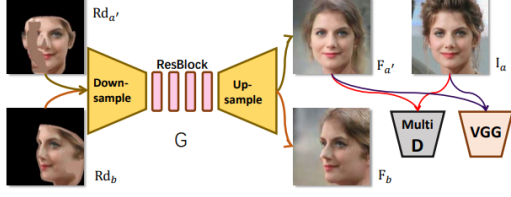


Figure 2. render-to-image module

The loss function of the discriminator includes the adversarial loss:

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_I [\log D(I_a)] + \mathbb{E}_{Rd} [\log (1 - D(G(Rd_{a'})))]$$

The feature loss is regularizing the distance from features of multiple layers of the discriminator between input and generated images.

$$\mathcal{L}_{FM}(G, D) = \frac{1}{N_D} \sum_{i=1}^{N_D} \left\| F_D^{(i)}(I_a) - F_D^{(i)}(G(Rd_{a'})) \right\|_1$$

Perceptual loss is achieved by using ImageNet pretrained VGG network with batch normalization. The equation is very similar to feature matching loss.

$$\mathcal{L}_{vgg}(G, D) = \frac{1}{N_{vgg}} \sum_{i=1}^{N_{vgg}} \left\| F_{vgg}^{(i)}(I_a) - F_{vgg}^{(i)}(G(Rd_{a'})) \right\|_1$$

In addition, to further improve image quality, we add the total variation loss [13] to remove artifacts in synthesized images.

$$\mathcal{L}_{tv} = \sum_{c=1}^C \sum_{w,h=1}^{W,H} \left| \hat{I}_{w+1,h,c}^f - \hat{I}_{w,h,c}^f \right| + \left| \hat{I}_{w,h+1,c}^f - \hat{I}_{w,h,c}^f \right|$$

The total objective function is:

$$\mathcal{L}_{total} = \mathcal{L}_{GAN} + \lambda_1 \mathcal{L}_{FM} + \lambda_2 \mathcal{L}_{vgg} + \lambda_3 \mathcal{L}_{tv}$$

4. Experiment and Result

4.1. Datasets

We use Face Transformer and Render and Rotation method discussed above to do face recognition and side face generation on two dataset, specifically FEI and CPLFW.

FEI[6] is a face dataset containing a set of face images taken between June 2005 and March 2006 at the Artificial Intelligence Laboratory of FEI in São Bernardo do Campo, São Paulo, Brazil contains, with 14 different pose images for 200 individuals. All images are colourful and taken against a white homogeneous background with profile rotation of up to 180 degrees. All faces are mainly represented by students and staff at FEI, between 19 and 40 years old with

distinct appearance, hairstyle, and adorns. The number of male and female subjects are exactly the same and equal to 100.

Cross-Pose LFW (CPLFW) [26] is a similar database with famous LFW dataset, both serves as an unconditional dataset for face recognition task. The difference of CPLFW comparing to LFW is that it has more pose variation of a single individual, and in the negative pair, age, gender and race are nearly the same, so those will not affect the accuracy. To simplify the training process, we select a subset of images to form the training set and testing set, with 2924 and 1106, respectively.

Relatively speaking, FEI dataset is easier for face recognition than CPLFW, since the background and clarity among images are well controlled. This phenomenon is also reflected by our experiment results.

4.2. Face Recognition

Firstly, we test the side face recognition accuracy among test databases. We use Face ViT pretrained on MS-Celeb-1M[8] dataset, which is a dataset of 10 million face images harvested from Internet for developing face recognition technologies. Since the training dataset is large enough, we do not implement finetune on test dataset, and the experiment results show that the ability of pretrained model is sufficient.

Table 1 shows that comparing to frontal face, recognition model on side face performs poorly on side face recognition. However, if the images in dataset are clean, the recognition model can also predict a great result, regardless of the rotation angle. But this kind of images can only be obtained by lab experiment. In real world, the model always needs to predict data like LFW and CPLFW, so we need to modify our recognition model to make it strong enough for those side face tasks. Besides, if we implement some pre-processing procedure for the CPLFW dataset, for example, alignment, we can get a even higher accuracy. However, after the alignment, it is hard to rotate, so, we use the original CPLFW data to do the recognition and the rotation.

4.3. Frontal Face Generation

To get the result of the frontal face generation model, we initially apply the model pretrained on **CASIA-WebFace**[24] and **MS-Celeb-1M**[8] to the dataset, from the Table3, we could see that the generation for Fei is quite good, the face and five organs are clear but with a little difference like no dark eye on the left side. The clear background and small domain gap might be the reason why the model could generate well. So we decide to finetune the model with the Fei dataset.

Table 1. Performance on LFW, CPLFW and FEI Database

Pretrained Data	Model	LFW	CPLFW	CPLFW with Preprocessing	FEI Rotation<30°	FEI Rotation>30°
MS-Celeb-1M	Face ViT	99.77	25.7	92.93	99.50	99.50

Table 2. The finetune results on Fei with the epoch increasing

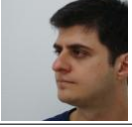

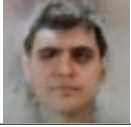
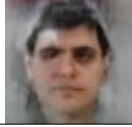
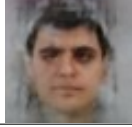
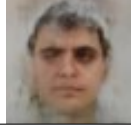


Original	25	50	75	100	125	150	200
							

Table 3. The output for each dataset using pretrained model


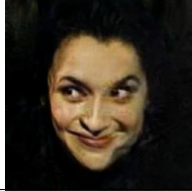
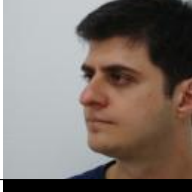
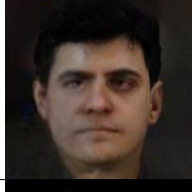


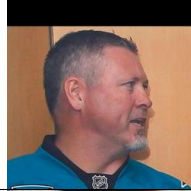

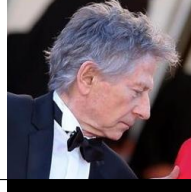
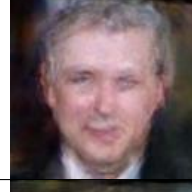


Dataset	Original	Generated
CelebA-HQ		
Fei		
CPLFW		

Table 4. The output for CPLFW on finetuned model

Original	Generated
	
	
	

But the results for CPLFW is not that good, we can not see clear face feature from the generated images, which may due to the large domain gap between the CPLFW and the original training set. So we train the model from the scratch.

The training process is done on 7 NVIDIA GeForce 2080 Tis with 11 Gb and 2 cards are used for rendering the the rest are for training. We set the feature matching loss parameter λ_1 and weight of vgg loss λ_2 to 10 and the weight of total variance loss λ_3 as 0.00001.

From the original pretrained model, we could see that some of the detail of the people is lossing, and we record the results with the epoch increasing as shown in Table 3, and we could see that the model learns some feature in detail such as the black eyes.

The finetuned model run on the test set is shown in Table4, from the result we could see the face structure but it is not that clear, and more likely to be a oil painting.

5. Analysis and Conclusion

From the result we could see that the frontal face generation model could perform well, and it could generate the face well for the Fei dataset. But the result on CPLFW is not as good as we expect. There might be following reasons why its result is not that good:

1. The training time is not enough, from the experiment we could see that with the finetuning epoch increasing, the generated images are more clear and the details are more reasonable and natural. The previoius work is done after thousands of epochs but ours could only be trained about 500 epochs. So the model might not learn enough from the training dataset.

2. The 3D model might not that accurate. The framework is based on the assumption that the 3D model for the face is quite accurate which might not be true for the unconditioned images. We use 3DFFA to generate the 3D fitting model, and it is clear that if the model estimate the 3D

model based only on single image, the output is likely to be not accurate. This explains why there are several identities with close generated frontal face.

3. The hyper-parameters are not fitting to the new dataset. We apply the similar hyper-parameters to that of the previous work due to the time limit, but these might not be fitting to our dataset. We think the result would improve if we could justify a better parameters.

4. The dataset is biased. The dataset we select is kind of biased, which contains more men images, which leads the result likely to generate the men's face. So generated women's face with short hair. And the background with a large variety, so it is hard to help the model to converge to optimal, and the poor background introduces noise to the model.

5. The assumption of the model is strict. The model suppose the rotation is only horizontal, but in practice, the image with yaw, pitch and roll, so the it hard to get corresponding pair if we still simply with only yaw angle. The illumination should also be considered. The work just directly project the 3D rendered model to 2D, but the projected image should consider the different illumination conditions, so that the generated images could be close to the natural one.

6. The alignment is not good. We try to do the alignment to make the image containing similar content, but the alignment for the image is not that good, so it is hard for the model to get the corresponding points between original and rotated images.

Therefore, there are several aspects to improve the performance: 1. A better generation model. We apply Pixel2pixel generation model to get the frontal image, but there are some recent models like StyleGan could perform better. 2. The model could be more fitting to the dataset. The model suppose that there are only one image to train, but in practice, there could be several images for one identity from different angles, which could be applied to have a complete view of the identity.

Overall, we build a face recognition model using Transformer which is widely used in natural language processing, and the recognition result is quite good in several benchmark datasets. And we also successfully build a frontal face generation model. Although the final result can not improve the recognition result based on our model, we recognize the reason of failure and the model also generates some reasonable output.

References

- [1] M. Abavisani and V. M. Patel. Deep multimodal subspace clustering networks. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1601–1614, 2018.
- [2] J. Cao, Y. Hu, H. Zhang, R. He, and Z. Sun. Learning a high fidelity pose invariant model for high-resolution face frontalization. *Advances in neural information processing systems*, 31, 2018.
- [3] J. Deng, S. Cheng, N. Xue, Y. Zhou, and S. Zafeiriou. Uv-gan: Adversarial facial uv map completion for pose-invariant face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7093–7102, 2018.
- [4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.
- [5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [6] V. do Amaral and C. E. Thomaz. Normalização espacial de imagens frontais de face. 2008.
- [7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.
- [8] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision – ECCV 2016*, pages 87–102, Cham, 2016. Springer International Publishing.
- [9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.
- [10] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.
- [11] R. Huang, S. Zhang, T. Li, and R. He. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *Proceedings of the IEEE international conference on computer vision*, pages 2439–2448, 2017.
- [12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.
- [13] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.
- [14] H. Kato, Y. Ushiku, and T. Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3907–3916, 2018.
- [15] C.-H. Lee, Z. Liu, L. Wu, and P. Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Pro-*

- ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5549–5558, 2020.
- [16] S. Liu and W. Deng. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 730–734, 2015.
 - [17] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song. Sphreface: Deep hypersphere embedding for face recognition. 04 2017.
 - [18] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019.
 - [19] A. Pumarola, A. Agudo, A. M. Martinez, A. Sanfeliu, and F. Moreno-Noguer. Ganimation: Anatomically-aware facial animation from a single image. In *Proceedings of the European conference on computer vision (ECCV)*, pages 818–833, 2018.
 - [20] S. I. Serengil and A. Ozpinar. Lightface: A hybrid deep face recognition framework. In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 23–27. IEEE, 2020.
 - [21] L. Tran, X. Yin, and X. Liu. Representation learning by rotating your faces. *IEEE transactions on pattern analysis and machine intelligence*, 41(12):3007–3021, 2018.
 - [22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.
 - [23] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.
 - [24] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.
 - [25] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker. Towards large-pose face frontalization in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3990–3999, 2017.
 - [26] T. Zheng and W. Deng. Cross-pose lfw: A database for studying cross-pose face recognition in unconstrained environments. Technical Report 18-01, Beijing University of Posts and Telecommunications, February 2018.
 - [27] Y. Zhong and W. Deng. Face transformer for recognition, 2021.
 - [28] H. Zhou, J. Liu, Z. Liu, Y. Liu, and X. Wang. Rotate-and-render: Unsupervised photorealistic face rotation from single-view images, 03 2020.
 - [29] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.