R^3 Adversarial Network for Cross Model Face Recognition

Ken Chen, Yichao Wu, Haoyu Qin, Ding Liang Xuebo Liu, Junjie Yan
Sensetime Group Limited

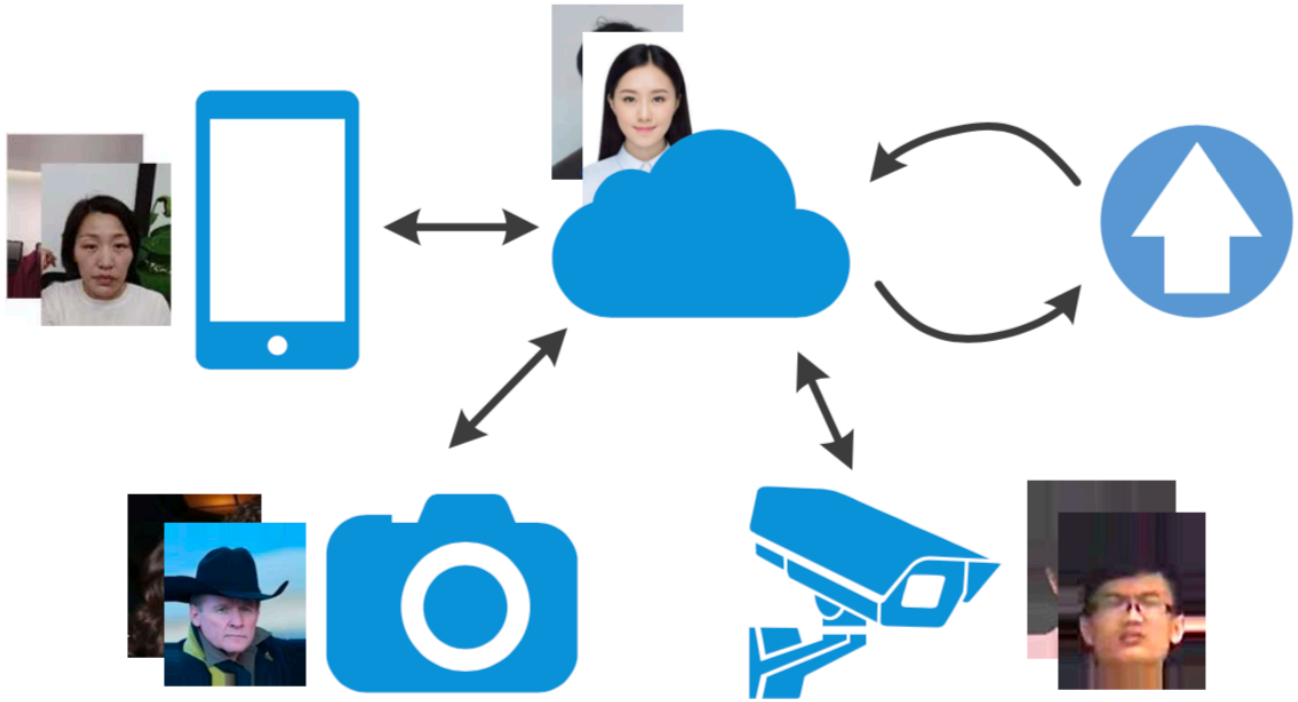
CVPR 2019

Sungman, Cho.

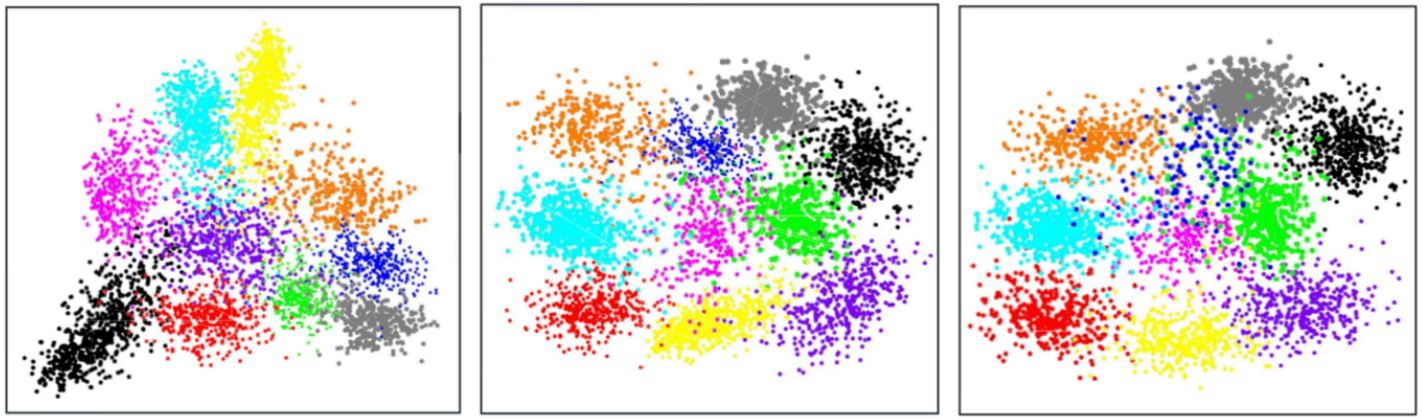
Introduction

Cross Model Face Recognition ?

- Pursuing interaction between information collected from various terminals is a new trend.



Cross Model Face Recognition ?



(a) (b) (c)

Figure 2: The feature distributions of two typical face recognition models.(a), (b) and (c) are feature distribution of source model, transformation model and target model, respectively.

Cross Model Face Recognition ?

- The core of this problem is to make **features extracted from different models comparable.**
- To solve this problem, from the perspective of Bayesian modeling, we propose $\mathrm{R}^3\mathrm{AN}$.
- $\mathrm{R}^3\mathrm{AN}$: Reconstruction, Representation and Regression.
- Introduce **adversarial learning into the reconstruction** path for better performance.

Why $\mathbf{R^3AN}$?

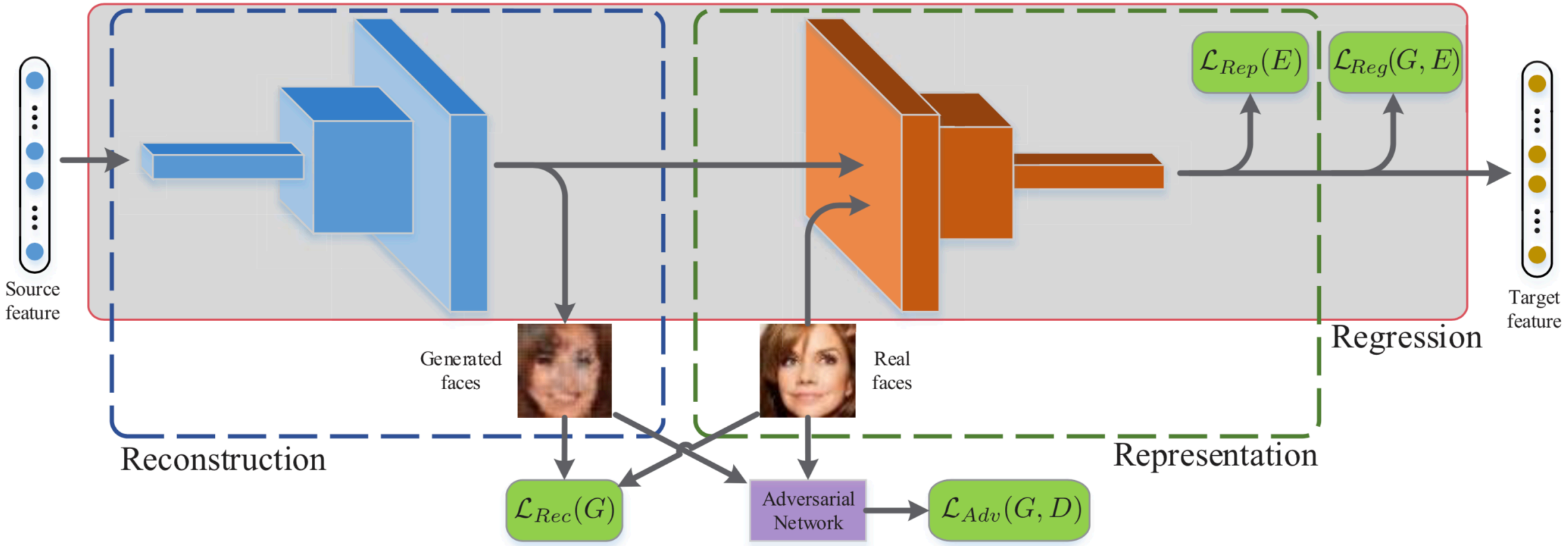
- Single model **lacks the ability** to achieve satisfactory performance when **various domains, applications** and response time requirement are taken into account.

- It violates **privacy policy in common sense**. Uploading and storing user's face images are generally forbidden in industrial community.

Contribution

- Raise the CMFR problem for the first time, which possesses considerable economic and social significance.
- $\mathbf{R^3AN}$ is super fast and valid when solving this problem
- Adversarial learning greatly improves the performance of $\mathbf{R^3AN}$ and recovers higher quality face image, which may **give valuable hints for improving the original face recognition models.**

Architecture



Basic Model

X, Y : source and target feature

Find a mapping function to **maximize the conditional probability $\boldsymbol{P(Y|X)}$.**

map X to Y in one-dimensional space directly by a naïve model like MLP.

→ yields unsatisfactory result.

Bayesian based Model

latent variables $h \in H = \{h_1, \ldots h_K\}$

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \sum_{h \in H} P(h|\boldsymbol{X}) P(\boldsymbol{Y}|\boldsymbol{X}, h).$$

X and Y is conditionally independent when h is given.

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \sum_{h \in H} P(h|\boldsymbol{X}) P(\boldsymbol{Y}|h).$$

h should be a latent variable independent of models.

$$P(\boldsymbol{Y}|\boldsymbol{X}) = P(\boldsymbol{I}|\boldsymbol{X}) P(\boldsymbol{Y}|\boldsymbol{I})$$

Bayesian based Model

latent variables $h \in H = \{h_1, \ldots h_K\}$

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \sum_{h \in H} P(h|\boldsymbol{X}) P(\boldsymbol{Y}|\boldsymbol{X}, h).$$

X and Y is conditionally independent when h is given.

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \sum_{h \in H} P(h|\boldsymbol{X}) P(\boldsymbol{Y}|h).$$

h should be a latent variable independent of models.

$$P(\boldsymbol{Y}|\boldsymbol{X}) = P(\boldsymbol{I}|\boldsymbol{X}) P(\boldsymbol{Y}|\boldsymbol{I})$$

$\mathbf{R^3AN}$

- **Reconstruction** : $P(I|X)$ → recover the original image.
- **Representation** : $P(Y|I)$ → extract feature from latent face images
- **Regression** : views reconstruction and representation as a unified problem. (jointly optimize the whole model)

Reconstruction Path

Restore original face images from extracted features, it can be regarded as generator (G)

Naïve Reconstruction

$$\mathcal{L}_{Rec}(G) = \mathbb{E}_{\boldsymbol{X},\boldsymbol{I}}\left[||\boldsymbol{I} - G\left(\boldsymbol{X}\right)||_2\right],$$

Adversarial Reconstruction

$$\mathcal{L}_{Adv}(G, D) = -\mathbb{E}_{\boldsymbol{X}}\left[\log\left(1 - D\left(G\left(\boldsymbol{X}\right)\right)\right)\right].$$

Representation Path

Representation path acts as an feature extractor, dented as E.

It takes the original face images as input and learns representation of the target model.

Representation path can be also considered as a knowledge distiller, which can transfer the knowledge of the target model to the feature extractor module.

$$\mathcal{L}_{Rep}(E) = \mathbb{E}_{\boldsymbol{I},\boldsymbol{Y}}\left[||\boldsymbol{Y} - E\left(\boldsymbol{I}\right)||_2\right].$$

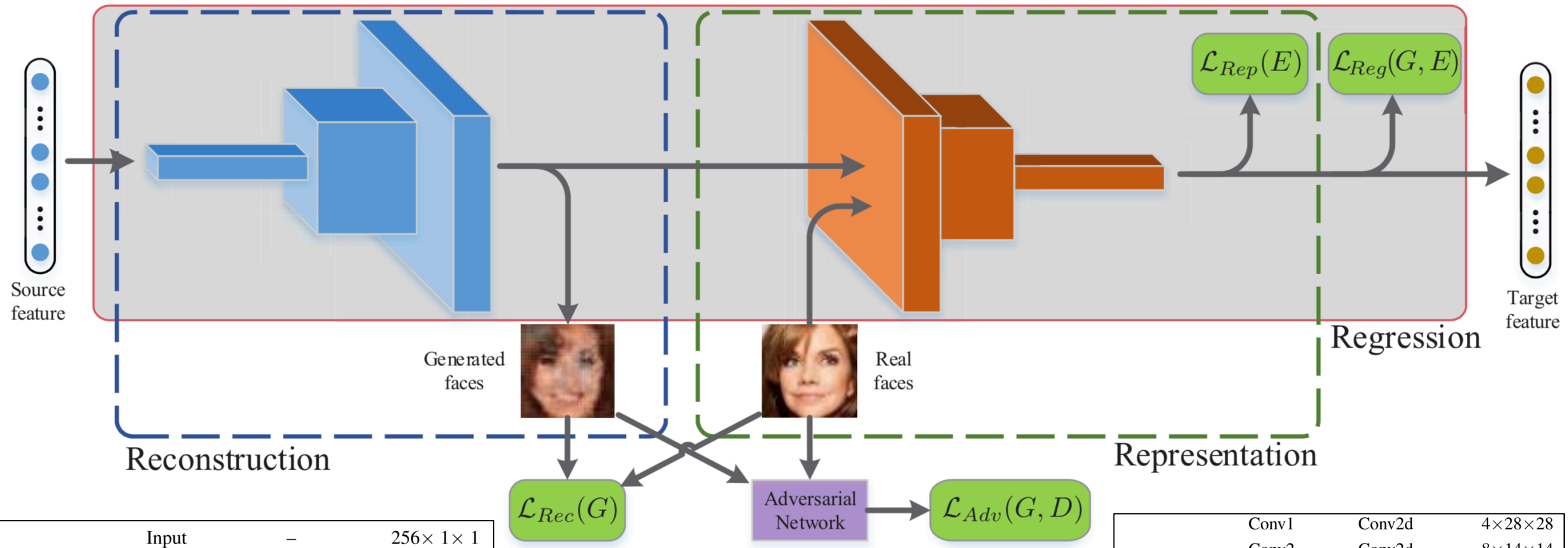
Regression Path

Combines reconstruction and representation together into a unified framework, and is used to jointly optimize the above two path.

$$\mathcal{L}_{Reg}(G, E) = \mathbb{E}_{\boldsymbol{X},\boldsymbol{Y}}\left[||\boldsymbol{Y} - E\left(G\left(\boldsymbol{X}\right)\right)||_2\right].$$

Architecture

Extractor	Conv1	bottleneck	32×28×28
	Conv2	bottleneck	64×14×14
	Conv3	bottleneck	96×14×14
	Conv4	bottleneck	160× 7× 7
	Conv5	bottleneck	320× 7× 7
	Conv6	Conv2d 1×1	1280× 7× 7
	Pooling	AvgPool2d	1280× 1× 1
	FC	Fully Connection	256× 1× 1



Generator	Input	–	256× 1× 1
	fConv1	ConvTranspose2d	512× 4× 4
	fConv2	ConvTranspose2d	128× 7× 7
	fConv3	ConvTranspose2d	32×14×14
	fConv4	ConvTranspose2d	8×28×28
	fConv5	ConvTranspose2d	3×56×56

Discriminator	Conv1	Conv2d	4×28×28
	Conv2	Conv2d	8×14×14
	Conv3	Conv2d	16×7×7
	Conv4	Conv2d	32×4×4
	Conv5	Conv2d	64×2×2
	Conv6	Conv2d	128×1×1
	FC	Fully Connection	1

Optimization

Final optimization goal :

$$(G^*,E^*) = \arg\min_{G,E}\max_{D}[\lambda_{Rec}\mathcal{L}_{Rec}(G) + \lambda_{Adv}\mathcal{L}_{Adv}(G,D) + \lambda_{Reg}\mathcal{L}_{Reg}(G,E)] + \lambda_{Rep}\mathcal{L}_{Rep}(E).$$

Optimization

Random initialize G, E and D
repeat
  for number of training epochs **do**
    for number of mini-batches **do**
      // for discriminator D
      $\theta_d \leftarrow \theta_d - \mu \frac{\partial \mathcal{L}_D(\boldsymbol{X},\boldsymbol{I};\theta_g,\theta_d)}{\partial \theta_d}$
      // for generator G
      $\theta_g \leftarrow \theta_g - \mu \frac{\partial \mathcal{L}_G(\boldsymbol{X},\boldsymbol{I};\theta_g,\theta_d)}{\partial \theta_g}$
      // for extractor E
      $\theta_e \leftarrow \theta_e - \mu \frac{\partial \mathcal{L}_{Rep}(\boldsymbol{I},\boldsymbol{Y};\theta_e)}{\partial \theta_e}$
      // for generator G and extractor E
      $(\theta_g, \theta_e) \leftarrow (\theta_g, \theta_e) - \mu \frac{\partial \mathcal{L}_{Reg}(\boldsymbol{X},\boldsymbol{Y};\theta_g,\theta_e)}{\partial (\theta_g,\theta_e)}$
    end for
  end for
until convergence, got $\hat{\theta}_g = \theta_g$, $\hat{\theta}_e = \theta_e$, $\hat{\theta}_d = \theta_d$
return $\hat{\boldsymbol{\theta}}_{\boldsymbol{g}}, \hat{\boldsymbol{\theta}}_{\boldsymbol{e}}, \hat{\boldsymbol{\theta}}_{\boldsymbol{d}}$

Experiments

Networks	Abbreviation	Top1 Acc
MobileNetV2(T=6) [23]	Mb-6	92.84
MobileNetV2(T=10) [23]	Mb-10	93.94
MobileNetV2(T=16) [23]	Mb-16	94.29
ResNet-50 [9]	Res50	97.48
ResNet-101 [9]	Res101	98.12
DenseNet121 [11]	Dns121	97.45
DenseNet161 [11]	Dns161	97.70
PolyNetE [32]	Poly	98.46

Table 2: Identification results of different models on MegaFace dataset. 'Top1 Acc' refers to the top-1 face identification accuracy rate with 1M distractors.

Experiments

Architecture	Rec Adv	Rec L2	Rep L2	Reg L2	Top1 Accuracy
FC	×	×	×	×	83.92
Arch1	×	✓	✓	×	85.65
Arch2	✓	×	✓	×	83.41
Arch3	×	×	×	✓	94.05
Arch4	×	✓	×	✓	94.21
Arch5	×	✓	✓	✓	94.93
Arch6	✓	×	×	✓	94.80
Arch7	×	×	✓	✓	NAN
R^3AN	✓	✓	✓	✓	**95.97**

Table 3: Identification results of CMFR between MobileNetV2 (T=6) and ResNet-101 based on different architectures. Each row in this table is an architecture, and each column means a specific training process. The '✓' and '×' represent for whether the architecture contains the process or not. 'Rec:Adv' and 'Rec:L2' means optimizing the generator by adversarial loss or L2 loss; 'Rep:L2' is extractor's optimization; 'Reg:L2' represents the regression path. 'Top1 Accuracy' refers to the top-1 face identification accuracy rate with 1M distractors.

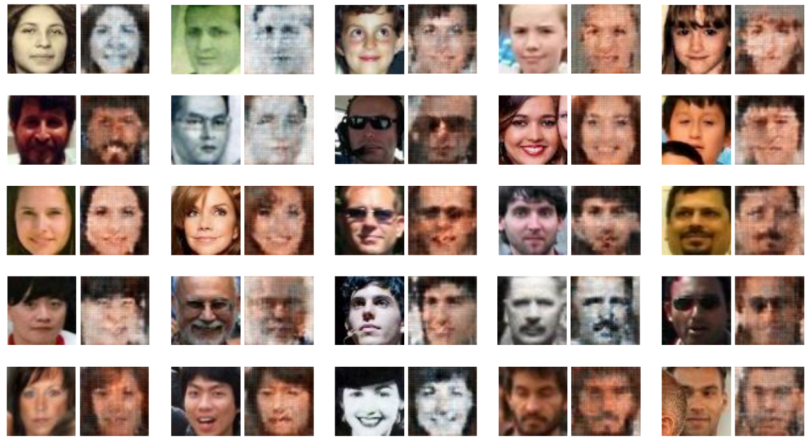
Source : MobileNet V2 → Target: ResNet-101

Experiments

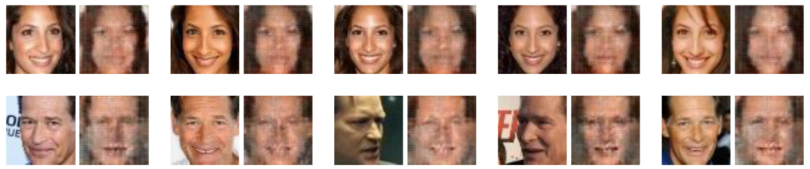
Src	Tgt	Src	Tgt	FC|Tgt	R^3AN|Tgt
Mb-6	Mb-10	92.84	93.94	83.66	94.36
Mb-10	Mb-6	93.94	92.84	84.17	94.69
Mb-6	Mb-16	92.84	94.29	83.54	94.18
Mb-10	Mb-16	93.94	94.29	84.19	94.33
Mb-6	Res50	92.84	97.48	83.78	94.48
Mb-6	Res101	92.84	98.12	83.92	95.97
Mb-6	Poly	92.84	98.46	82.92	97.44
Mb-6	Dns161	92.84	97.70	82.95	95.66
Res50	Mb-6	97.48	92.84	87.49	97.60
Res101	Mb-6	98.12	92.84	88.17	98.19
Poly	Mb-6	98.46	92.84	88.51	98.34
Dns161	Mb-6	97.70	92.84	87.63	97.64
Res50	Res101	97.48	98.12	89.29	97.69
Res101	Res50	98.12	97.48	89.38	97.86
Dns121	Dns161	97.45	97.70	87.41	97.81
Res50	Poly	97.48	98.46	87.24	98.29

Table 5: Results of CMFR among different prior models. We use proposed models to map distribution of 'Src' (source model) to the distribution 'Tgt' (target model). The evaluation is established by taking the learned representation from the left model of '|' as probe and right model's output as gallery. 'FC' and 'R^3AN' refers to the architecture in the first and last row of Tab. 3. Results are the top-1 face identification accuracy rates with 1M distractors.

Visualization

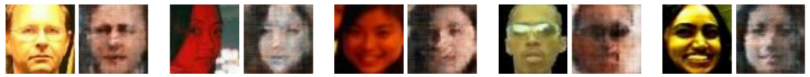


(a) Normal



(b) Same person

(c) Occlusion



(d) Lighting

Figure 4: The visualization of real faces and generated images from generator. Real faces are on the left, and generated images are on the right.

Visualization

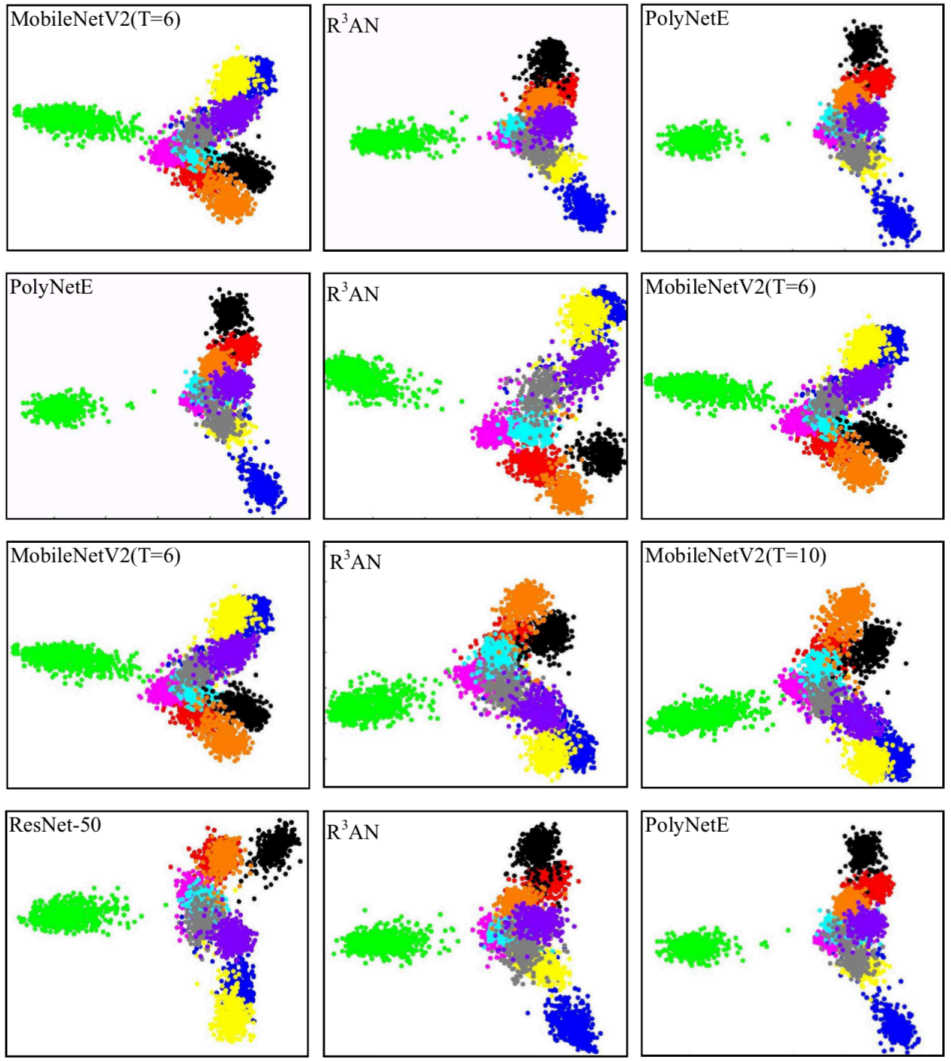


Figure 5: The visualization of feature distribution. From left to right in each row, images of distribution are from source model, R^3AN and target model, respectively.

Thanks.

☺