Supplement: Feature Transfer Learning for Face Recognition with Under-Represented Data

Xi Yin[†][*], Xiang Yu[‡], Kihyuk Sohn[‡], Xiaoming Liu[†] and Manmohan Chandraker[§][‡]
[†]Michigan State University
[‡] NEC Laboratories America
[§]University of California, San Diego
{yinxi1,liuxm}@cse.msu.edu, {xiangyu,ksohn,manu}@nec-labs.com

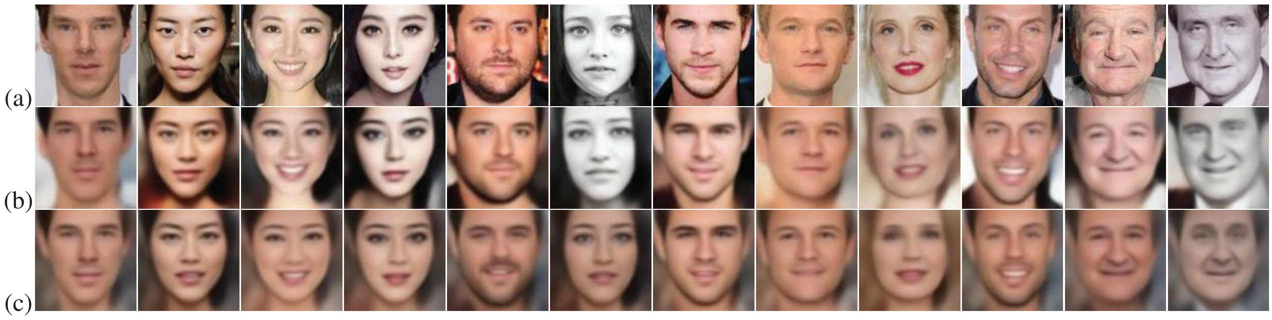


Figure 1: Similarity visualization of closest-to-center sample, its reconstruction and the feature center reconstruction. (a) Sample image whose feature is closest to the feature center. (b) The reconstructed image of (a). (c) The reconstructed image from the class feature center. We observe that the feature center corresponds to a neutral frontal face, which also shares similarity to the reconstructed image of the closest sample (row (b)).

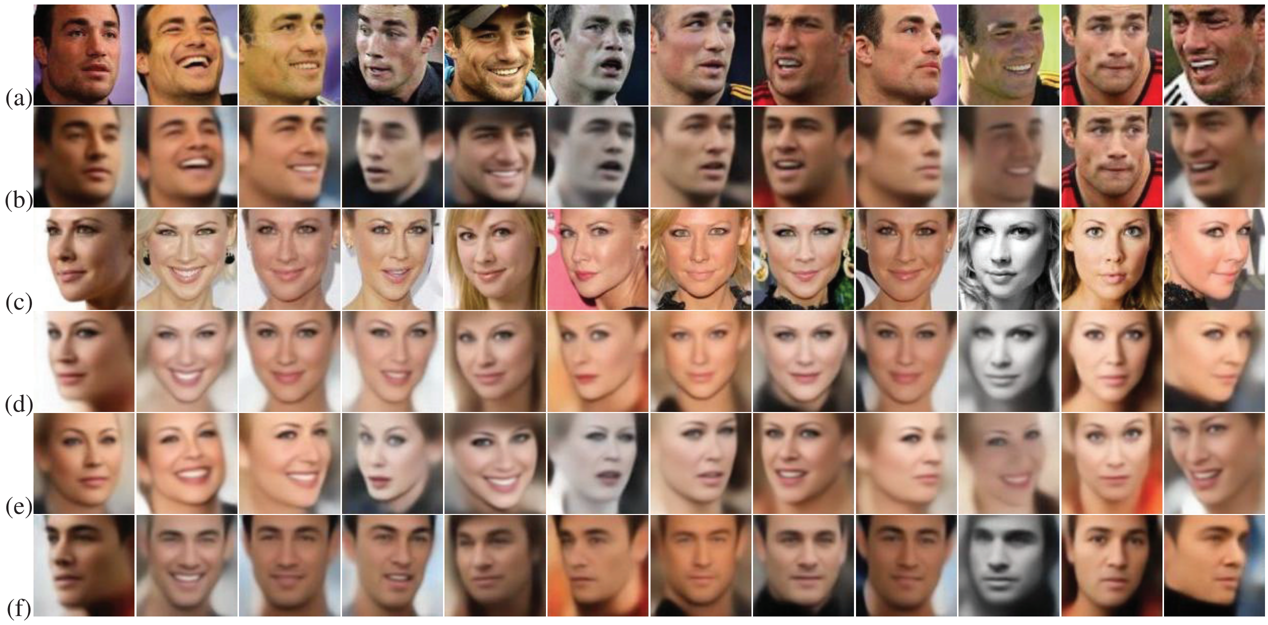


Figure 2: Feature transfer visualization between two classes. (a) Original images of class 1: $\mathbf{x}_1$. (b) Reconstructed images of (a): $\mathbf{x}'_1$ (c) Original images of class 2: $\mathbf{x}_2$. (d) Reconstructed images of (c): $\mathbf{x}'_2$. (e) Reconstructed images of the transferred features from class 1 to class 2: $\mathbf{x}'_{12}$. (f) Reconstructed images of the transferred features from class 2 to class 1: $\mathbf{x}'_{21}$. It is clear that the transferred features share the same identity as the target class while keep the source image's intra-class variance including pose, expression, illumination, *etc*. It shows the effectiveness of the proposed feature transfer in enlarging the intra-class variance for *UR* classes.

1. Feature Visualization

Center Visualization Given multiple images of the same class, we extract features using Enc and compute a feature center. We apply Dec on the feature center to reconstruct a center face image. We also find one sample that is closest to the center in the feature space and decode the sample features. Figure 1 shows several examples. Each row shows the original image of the closest sample, the reconstructed image from this sample, the reconstructed image of the feature center. We observe that the feature centers correspond to frontal neutral faces, which are visually similar to the reconstructed images from the closest samples. In some cases, the feature center shows a smiling face, which happens when the majority of the images in this class is smiling.

Feature Transfer We perform feature transfer in the feature space $\mathbf{g}$. The transferred features can be visualized using Dec. Let $\mathbf{x}_{1,2}$, $\mathbf{x}'_{1,2}$, $\mathbf{g}_{1,2}$, $\mathbf{c}_{1,2}$ denote the input images, reconstructed images, encoded features, and feature centers of two classes, respectively. Let $\mathbf{Q}$ denote the PCA basis of the intra-class variance. We transfer features from class 1 to class 2: $\mathbf{g}_{12} = \mathbf{c}_2 + \mathbf{Q}\mathbf{Q}^T(\mathbf{g}_1 - \mathbf{c}_1)$, and visualize the decoded images as $\mathbf{x}'_{12}$. We also transfer features from class 2 to class 1: $\mathbf{g}_{21} = \mathbf{c}_1 + \mathbf{Q}\mathbf{Q}^T(\mathbf{g}_2 - \mathbf{c}_2)$, and visualize the decoded images as $\mathbf{x}'_{21}$.

Figure 2 shows the examples of feature transfer between two classes. The proposed feature transfer succeeded in transferring the source class's intra-class variance onto the target class's center. The visualizations of the transferred features consistently preserve the target class's identity via incorporating the source image's attributes, *i.e.* pose, expression, lighting condition, *etc*., which shows that our feature transfer is effective in enlarging the intra-class variance.

PCA Basis In our framework, we use PCA to capture the intra-class variance. Here we visualize what is being captured in each basis. Specifically, we add one basis to the feature center of one class to generate a new feature representation of that class via $\mathbf{g}_i = \mathbf{c}_i + \mathbf{Q}(:, k) \cdot 0.1$, where $\mathbf{c}_i$ is the center of class i, $\mathbf{Q}(:, k)$ is the kth PCA basis, and 0.1 is the mean absolute coefficient of all images when projecting to the top 10 basis.

Figure 3 shows the results of several examples. It is clear that each PCA basis consistently captures a mixture of pose, expression, illumination variations. For example, adding the 1st basis improves the image quality with good lighting condition; adding the 6th basis turns the face to left and makes it smile; adding the 7th basis turns the face downward slightly and opens the mouth; and *etc*. It is critical that the PCA basis captures the various intra-class variations so that the feature transfer is semantically meaningful. This visualization supports that the reconstruction task in our baseline framework encourages the feature space $\mathbf{g}$ to capture these variations.

Feature Interpolation The interpolation between two face representations can show the transition from one to the other. This visualization is widely used in GAN-based frameworks [5,6]. However, previous work visualize this transition with a mixed change of identity and non-identity variations. In our approach, we model face as a linear combination of center and intra-class variance. Therefore, we can separate the visualization into two parts. Let $\mathbf{g}_{1,2}$, $\mathbf{c}_{1,2}$ denote the encoded features and the feature centers of two samples from different classes respectively. Previous work generates a new representation as $\mathbf{g} = \mathbf{g}_1 + (\mathbf{g}_2 - \mathbf{g}_1) * \alpha$. In our work, we can generate a smooth transition of non-identity change as $\mathbf{g} = \mathbf{c}_1 + \mathbf{Q}\mathbf{Q}^T(\mathbf{g}_2 - \mathbf{c}_2) * \alpha$, which is the same as the proposed feature transfer when $\alpha = 1$. On the other hand, we can also generate a smooth transition of identity change as $\mathbf{g} = \mathbf{g}_1 + (\mathbf{c}_2 - \mathbf{c}_1) * \alpha$. We vary α from 0.1 to 1 and visualize the transition results.

Figure 4 shows the above 3 transitions of 4 examples. Traditional method shows the change of identity and non-identity components simultaneously. In our approach, we can visualize two separate transitions for identity and non-identity changes. For example, (a) shows the transition from a male with slightly left pose to a female with right pose. Second row shows the same identity while the mouth is gradually open and the face is turning to the right at the same time. Third row shows the same attributes as the left image (pose and expression) while the identity is gradually changed to that of the right image. When the source and target image have opposite pose (example (d)), traditional interpolation generates undesirable artifacts. However, our method shows smooth transitions without any artifacts.

2. Ablation Study

In this paper, we propose a two-stage alternative training scheme to correct the classifier bias and learn a more discriminative feature representation. In the main manuscript, we have shown sufficient experimental results to support that we can learn a better feature representation, which is essential for face recognition. Here we present evidence to show that we have corrected the classifier bias as well.

In the ablation study, we perform classification on the hold-out testing set from MS-celeb-1M. We compare two methods. (1) FC: use the trained FC as the classifier. (2) NN: use the training images to calculate the class feature center as the gallery and use Nearest Neighbor for classification. It is observed in [1] that the L_2 norm of the weights for a *UR* class is smaller than that of a regular class, which suggests that the classifier is biased. Motivated by [1], we compare the weight norm of regular and *UR* classes, which can quantify the classifier bias.

The results are shown in Table 1. When using FC on the baseline models, the accuracy on the *UR* classes is very low. However when NN is used, the accuracy on the regular classes drops slightly while the accuracy on the *UR* classes improves significantly. This suggests that an *UR* sample that is closer to its center can be mis-classified into a neighboring regular class when using the trained FC. This bias is also consistent with the imbalance between the weight norm of regular and *UR* classes. After applying our FTL, we observe

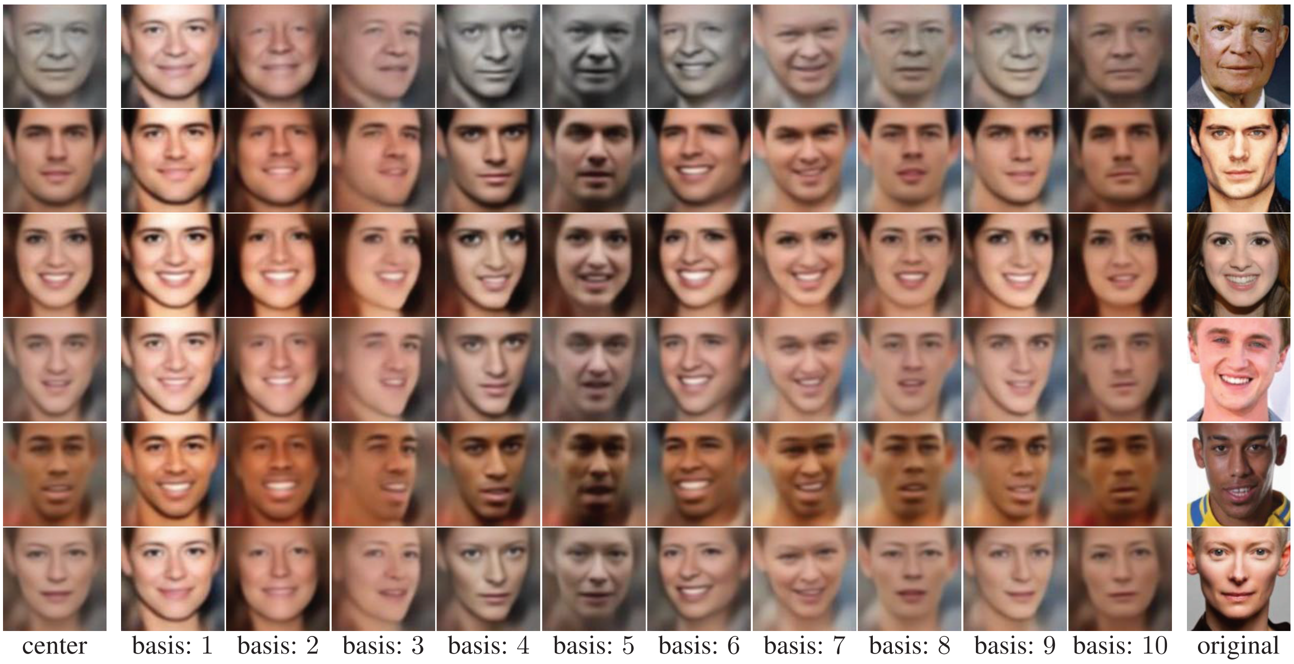


Figure 3: Visualization of the PCA basis. Column 1 shows the feature center reconstructed images. Column 2-11 shows the reconstructed image of adding one of the top 10 principal components to the centers. Column 12 shows one sample image from the corresponding class. We observe that each principal component captures a mixture of pose, expression, and illumination. For example, adding the 6th basis to the centers turns the faces to left and with smiling expression. Therefore the proposed baseline framework with reconstruction task encourage the intra-class variance being captured in features $\mathbf{g}$.

Test →		MS1M: FC		MS1M: NN		Weight Norm	
Train↓	Method↓	Regular	*UR*	Regular	*UR*	Regular	*UR*
10K0K	sfmx+m-L_2	92.03	–	90.21	84.64	0.427	–
10K10K	sfmx+m-L_2	90.76	0.15	89.48	84.10	0.430	0.126
	FTL (Ours)	95.18	88.32	92.27	88.16	0.379	0.356
10K30K	sfmx+m-L_2	93.59	2.04	90.60	86.40	0.401	0.139
	FTL (Ours)	96.26	81.89	91.76	88.72	0.366	0.270
10K50K	sfmx+m-L_2	93.73	4.76	90.24	87.11	0.387	0.141
	FTL (Ours)	96.66	68.52	92.08	89.36	0.352	0.235
10K0K	sfmx+m-L_2	94.07	93.15	93.68	93.46	0.291	0.281

Table 1: Additional results on the controlled experiments by varying the ratio between regular and *UR* classes in the training sets. Evaluation is done on the hold out test set from MS1M.

a consistent improvement when using FC for classification. The imbalance between the weight norm is also reduced to a large extent. However, the bias is more difficult to correct as the number of *UR* classes increases. In conclusion, the proposed FTL method is very effective in correcting the classifier bias and learning more discriminative features.

3. Data Distribution

Dataset Statistics Figure 6 shows the distribution of two public face datasets: CASIA-Webface [7] and the cleaned version of MS-celeb-1M [2]. Considering a class with no more than 20 images as an *UR* class, the specific statistics of regular and *UR* classes are shown in Table 3. There is a large portion of *UR* classes for both datasets, which only contributes a small amount of images to the full dataset. Training with such *UR* data will cause the classifier bias problem and lead to an inferior feature representation.

Dataset Distribution The main purpose of FTL is to enrich the biased distribution of an *UR* class. To visualize this effect, we project the features $\mathbf{f}$ onto 2D space. Figure 5 shows one example of the sample distribution before and after the feature transfer. Figure 5 (a) shows one regular class with a balanced distribution. When we select 20 images to form an *UR* class, the distribution is biased (Figure 5 (b)). After the proposed FTL, we observe that the distribution is enriched after transferring 100 and 300 samples from other regular classes, as shown in Figure 5 (c) and (d) respectively.

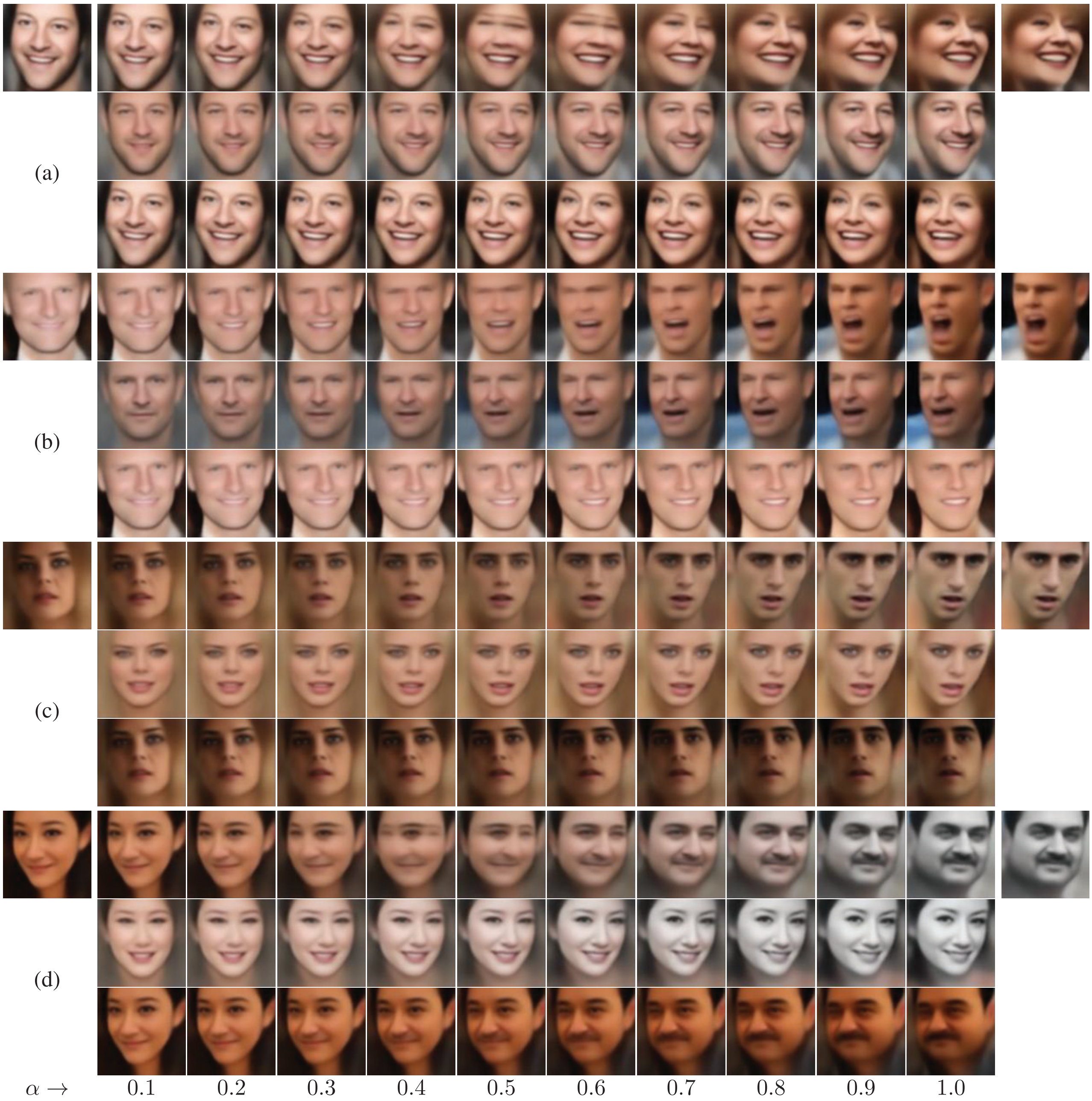


Figure 4: Transition from top-left image to top-right image via feature interpolation. For each example: first row shows the results of $\mathbf{g} = \mathbf{g}_1 + (\mathbf{g}_2 - \mathbf{g}_1) * \alpha$; second row shows the results of $\mathbf{g} = \mathbf{c}_1 + \mathbf{Q}\mathbf{Q}^T(\mathbf{g}_2 - \mathbf{c}_2) * \alpha$; third row shows the results of $\mathbf{g} = \mathbf{g}_1 + (\mathbf{c}_2 - \mathbf{c}_1) * \alpha$. While traditional interpolation shows a mixture change of identity and non-identity variations (first row), our approach can separate the interpolation for non-identity (second row) and identity (third row) changes.

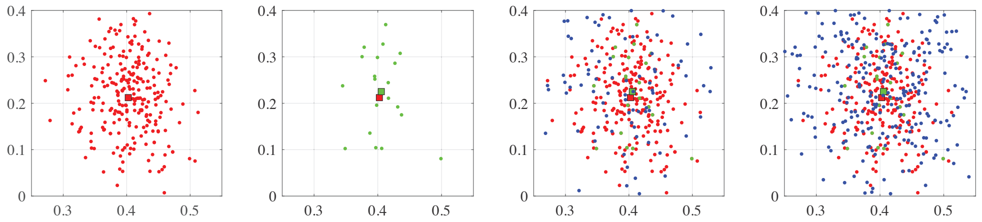
4. Network Structures

Table 2 shows the network structures of our framework, which consists of an encoder Enc, a decoder Dec, a distillation network R, and an FC classifier. The Enc takes an input image $\mathbf{x} \in \mathbb{R}^{100\times100\times3}$ and generates a feature vector $\mathbf{g} \in \mathbb{R}^{320\times1}$. The Dec takes $\mathbf{g}$ as input and reconstructs the original input image as $\mathbf{x}' \in \mathbb{R}^{100\times100\times3}$. The R takes the features $\mathbf{g}$ as input and distills the intra-class variance to generate a more discriminative representation $\mathbf{f} \in \mathbb{R}^{320\times1}$. The FC takes $\mathbf{f}$ as input for classification with a linear layer, which is eliminated from the table. Batch Normalization [3] and ReLU [4] are used after each convolutional (Conv) and full convolutional (Fconv) layer except “Conv53”.

It is well known that adding skip connections between the encoder and the decoder helps to improve the visual

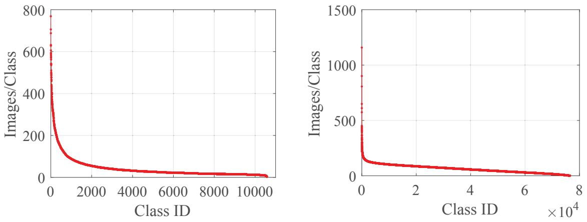
Enc			*Dec*			*R*		
Layer	Filter/Stride/Pad	Output Size	Layer	Filter/Stride/Pad	Output Size	Layer	Filter/Stride/Pad	Output Size
			FC		6 × 6 × 320	FC		6 × 6 × 320
Conv11	3 × 3/1/1	100 × 100 × 32	FConv52	3 × 3/1/1	6 × 6 × 160	FConv52	3 × 3/1/1	6 × 6 × 160
Conv12	3 × 3/1/1	100 × 100 × 64	FConv51	3 × 3/1/1	6 × 6 × 256	FConv51	3 × 3/1/1	6 × 6 × 256
Conv21	3 × 3/2/1	50 × 50 × 64	FConv43	3 × 3/2/1	12 × 12 × 256	-	-	-
Conv22	3 × 3/1/1	50 × 50 × 64	FConv42	3 × 3/1/1	12 × 12 × 128	-	-	-
Conv23	3 × 3/1/1	50 × 50 × 128	FConv41	3 × 3/1/1	12 × 12 × 192	-	-	-
Conv31	3 × 3/2/1	25 × 25 × 128	FConv33	3 × 3/2/1	24 × 24 × 192	-	-	-
Conv32	3 × 3/1/1	25 × 25 × 96	FConv32	3 × 3/1/1	24 × 24 × 96	-	-	-
Conv33	3 × 3/1/1	25 × 25 × 192	FConv31	3 × 3/1/1	24 × 24 × 128	-	-	-
Conv41	3 × 3/2/0	12 × 12 × 192	FConv23	3 × 3/2/1	48 × 48 × 128	-	-	-
Conv42	3 × 3/1/1	12 × 12 × 128	FConv22	3 × 3/1/1	48 × 48 × 64	-	-	-
Conv43	3 × 3/1/1	12 × 12 × 256	FConv21	3 × 3/1/0	50 × 50 × 64	-	-	-
Conv51	3 × 3/2/1	6 × 6 × 256	FConv13	3 × 3/2/1	100 × 100 × 64	-	-	-
Conv52	3 × 3/1/1	6 × 6 × 160	FConv12	3 × 3/1/1	100 × 100 × 32	Conv52	3 × 3/1/1	6 × 6 × 160
Conv53	3 × 3/1/1	6 × 6 × 320	FConv11	3 × 3/1/1	100 × 100 × 3	Conv53	3 × 3/1/1	6 × 6 × 320
AvgPool	6 × 6/1/0	1 × 1 × 320				AvgPool	6 × 6/1/0	1 × 1 × 320

Table 2: Network structures of different modules in the proposed framework.



(a) (b) (c) (d)

Figure 5: (a) The original distribution with center annotated as red box. (b) The faked *UR* distribution with a subset of samples and an estimated center (green box). (c) The enriched distribution after transferring 100 samples (blue) to this class. (d) The enriched distribution after transferring 300 samples (blue) to this class.



(a) CASIA-Webface (b) MS-celeb-1M

Figure 6: Number of images per class *vs*. class ID on CASIA-Webface and MS-celeb-1M show that the public large-scale face benchmarks have severe portion of UR data.

	CASIA-Webface		MS-celeb-1M	
	#Classes	#Images	#Classes	#Images
Full	10.6K	455.6K	76.5K	4753.3K
Regular	6.5K	393.1K	67.0K	4638.4K
UR	4.0K	62.5K	9.5K	114.9K

Table 3: Statistics of CASIA-Webface and MS-celeb-1M.

quality of the decoded images [8]. However, we do not add any skip connections to encourage the encoded features $\mathbf{f}$ to include more intra-class variance that is needed for image reconstruction by itself. This intra-class variance is important when performing feature transfer.

References

[1] Y. Guo and L. Zhang. One-shot face recognition by promoting underrepresented classes. *arXiv preprint arXiv:1707.05574*, 2017.

[2] Y. Guo, L. Zhang, Y. Hu, X. He, and J. Gao. MS-Celeb-1M: A dataset and benchmark for large scale face recognition. In *ECCV*, 2016.

[3] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[4] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[5] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

[6] L. Tran, X. Yin, and X. Liu. Disentangled representation learning GAN for pose-invariant face recognition. In *CVPR*, 2017.

[7] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint:1411.7923*, 2014.

[8] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker. Towards large-pose face frontalization in the wild. In *ICCV*, 2017.