Boosting Unconstrained Face Recognition with Auxiliary Unlabeled Data (Supplenmentary Material)

Yichun Shi Anil K. Jain

Michigan State University

shiyichu@msu.edu, jain@cse.msu.edu

A. Numerical Results on IJB-B, IJB-C

In Table 1 and Table 2, we show more numerical results on the IJB-C and IJB-B dataset, respectively. Since all the baseline methods (from other papers) are trained on different number of labeled images, we report the performance of our models trained on different labeled subsets for a more fair comparison. From the tables, we could observe that our models outperform most of the baselines with equal or less than 2M labeled data.

Method	Data	Model	Verification				Identification	
			1e-7	1e-6	1e-5	1e-4	Rank1	Rank5
Cao et al. [2]	13.3M	SE-ResNet-50	-	-	76.8	86.2	91.4	95.1
PFE [8]	4.4M	ResNet-64	-	-	89.64	93.25	95.49	97.17
ArcFace [3]	5.8M	ResNet-50	67.40	80.52	88.36	92.52	93.26	95.33
Ranjan et al. [7]	5.6M	ResNet-101	67.4	76.4	86.2	91.9	94.6	97.5
AFRN [6]	3.1M	ResNet-101	-	-	88.3	93.0	**95.7**	**97.6**
Baseline	500K	ResNet-50	51.13	66.44	77.58	87.73	90.90	94.50
Proposed	500K+70K	ResNet-50	60.33	71.24	80.31	88.18	91.81	94.96
Baseline	1.0M	ResNet-50	59.53	77.70	86.16	92.13	93.62	95.93
Proposed	1.0M+70K	ResNet-50	61.87	79.76	87.16	92.39	94.19	96.30
Baseline	2.0M	ResNet-50	67.64	78.66	88.16	93.48	94.34	96.34
Proposed	2.0M+70K	ResNet-50	**78.62**	84.91	90.61	93.77	95.04	96.80
Baseline	3.0M	ResNet-50	62.65	79.20	89.20	94.20	94.76	96.49
Proposed	3.0M+70K	ResNet-50	78.38	85.91	91.56	94.48	95.51	97.04
Baseline	3.9M	ResNet-50	62.90	82.94	90.73	94.57	94.90	96.77
Proposed	3.9M+70K	ResNet-50	77.39	**87.92**	**91.86**	**94.66**	95.61	97.13

Table 1: Performance comparison with state-of-the-art methods on the IJB-C dataset.

B. Architecture of Augmentation Network

The architecture of our augmentation network is based on MUNIT [5]. Let `c5s1-k` be a 5×5 convolutional layer with k filters and stride 1. `dk-IN` denotes a 3×3 convolutional layer with k filters and dilation 2, where IN means Instance Normalization [9]. Similarly, AdaIN means Adaptive Instance Normalization [4] and LN denotes Layer Normalization [1]. `fc8` denotes a fully connected layer with 8 filters. `avgpool` denotes a global average pooling layer. No normalization is used in the style encoder. We use Leaky ReLU with slope 0.2 in the discriminator and ReLU activation everywhere else. The architectures of different modules are as follows:

Method	Data	Model	Verification				Identification	
			1e-6	1e-5	1e-4	1e-3	Rank1	Rank5
Cao et al. [2]	13.3M	SE-ResNet-50	-	70.5	83.1	90.8	90.2	94.6
Comparator [10]	3.3M	ResNet-50	-	-	84.9	93.7	-	-
ArcFace [3]	5.8M	ResNet-50	40.77	84.28	91.66	94.81	92.95	95.60
Ranjan et al. [7]	5.6M	ResNet-101	**48.4**	80.4	89.8	94.4	93.3	96.6
AFRN [6]	3.1M	ResNet-101	-	77.1	88.5	94.9	**97.3**	**97.6**
Baseline	500K	ResNet-50	39.35	71.14	84.37	92.12	89.74	94.16
Proposed	500K+70K	ResNet-50	45.39	72.35	84.75	92.00	90.46	94.42
Baseline	1.0M	ResNet-50	45.75	80.11	90.19	94.48	92.37	95.78
Proposed	1.0M+70K	ResNet-50	41.59	82.10	90.09	94.64	92.88	95.91
Baseline	2.0M	ResNet-50	47.62	82.30	91.82	95.46	93.25	96.05
Proposed	2.0M+70K	ResNet-50	44.76	86.26	91.92	95.27	94.01	96.23
Baseline	3.0M	ResNet-50	42.77	82.86	92.48	95.78	93.80	96.23
Proposed	3.0M+70K	ResNet-50	43.09	87.31	**92.80**	95.70	94.35	96.53
Baseline	3.9M	ResNet-50	40.12	84.38	92.79	**95.90**	93.85	96.55
Proposed	3.9M+70K	ResNet-50	43.38	**88.19**	92.78	95.86	94.62	96.72

Table 2: Performance comparison with state-of-the-art methods on the IJB-B dataset.

- Style Encoder:
  `c5s1-32,c3s2-64,c3s2-128,avgpool,fc8`
- Generator:
  `c5s1-32-IN,d32-IN,d32-AdaIN,d32-LN, d32-LN,c5s1-3`
- Discriminator:
  `c5s1-32,c3s2-64,c3s2-128`

The length of the latent style code is set to 8. A style decoder (multi-layer perceptron) has two hidden fully connected layers of 128 filters without normalization, which transforms the latent style code to the parameters of the AdaIN layer.

C. Ablation over the Settings of Augmentation Network

In this section, we ablate over the training modules of the augmentation network. In particular, we consider to remove the following modules for different variants: Latent-style code for multi-mode generation (MM), Image Discriminator (D_I), Reconstruction Loss (Rec), Style Discriminator (D_z) and the architecture without downsampling (ND).

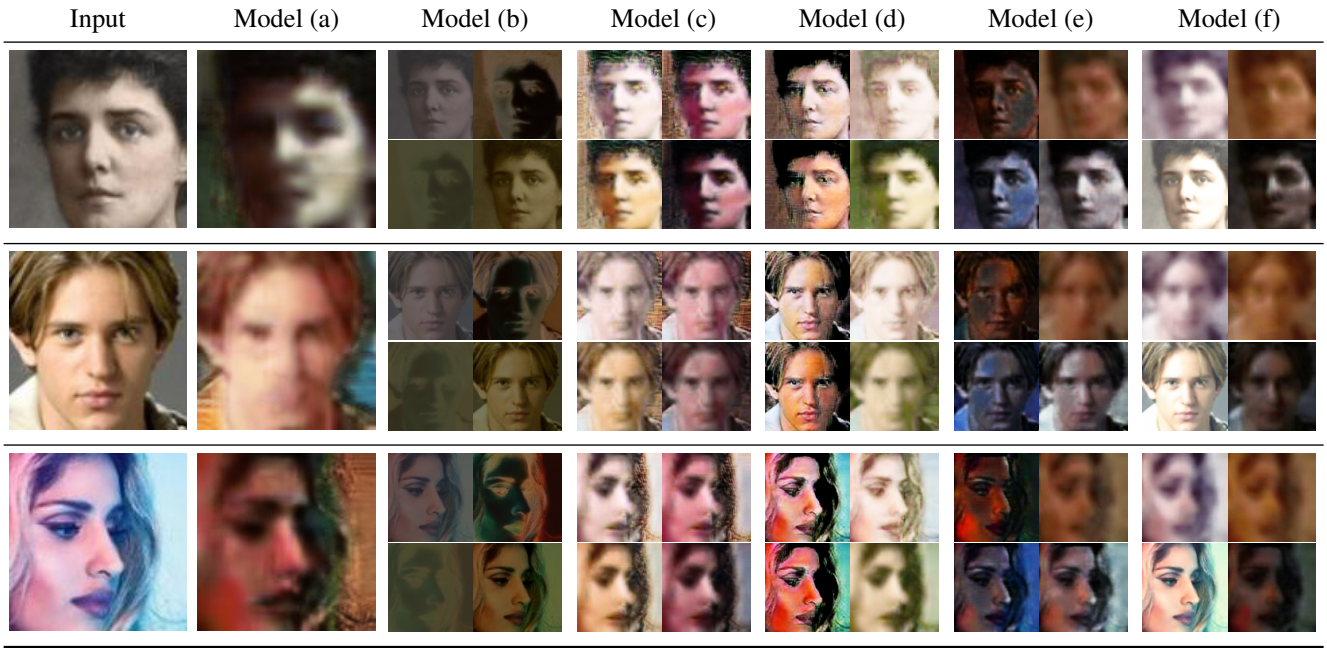

Figure 1: Ablation study of the augmentation network. Input images are shown in the first column. The subsequent columns show the results of different models trained without a certain module or loss. The texture style codes are randomly sampled from the normal distribution.

Model	Modules					IJB-C (Vrf)			IJB-C (Idt)		IJB-S (V2S)		LFW
	MM	D_I	Rec	D_Z	ND	1e-7	1e-6	1e-5	Rank1	Rank5	Rank1	Rank5	Accuracy
						72.74	85.33	90.52	94.99	96.75	56.35	66.77	99.82
(a)		✓			✓	74.80	87.58	91.94	95.51	97.09	56.98	65.66	99.80
(b)	✓		✓	✓	✓	75.32	88.00	91.71	95.42	97.04	57.54	66.72	99.75
(c)	✓	✓			✓	74.51	87.49	91.97	95.61	97.18	57.17	66.24	99.78
(d)	✓	✓	✓		✓	75.07	88.11	92.19	95.66	97.12	56.85	64.87	99.78
(e)	✓	✓	✓	✓		73.99	86.52	91.33	95.33	97.04	58.47	66.00	99.73
(f)	✓	✓	✓	✓	✓	77.39	87.92	91.86	95.61	97.13	57.33	65.37	99.75

Table 3: Ablation study over different training methods of the augmentation network. "MM", "D_I", "D_Z", "rec", "ND" refer to "Multi-mode", "Image Discriminator", "Reconstruction Loss", "Latent Style Discriminator" and "No Downsampling", respectively. The first row is a baseline that uses only the domain adversarial loss but no augmentation network. "Model (a)" is a single-mode translation network that does not use latent style code.

The qualitative results of different models are shown in Fig. 1. Without the latent style code (Model a), the augmentation network can only output one deterministic image for each input, which mainly applies blurring to the input image. Without the image adversarial loss (Model b), the model cannot capture the realistic variations in the unlabeled dataset and the style code can only change the color channel in this case. Without the Reconstruction Loss (Model c), the model is trained only with adversarial loss but without the regularization of content preservation. And therefore, we see clear artifacts on the output images. However, adding reconstruction loss alone hardly helps, since the latent code used in the reconstruction of the unlabeled images could be very different from the prior distribution $p(z)$ that we use for generation. Therefore, similar artifacts can be observed if we do not add latent code adversarial loss (Model d). As for the architecture, if we choose to use an encoder-decoder style network as in the original MUNIT [5], with downsampling and upsampling (Model e), we observe that the output images are always blurred due to the loss of spatial information. In contrast, with our architecture (Model f), the network is capable of augmenting images with diverse color, blurring and illumination styles but without clear artifacts.

Furthermore, we incorporate these different variants of augmentation networks into training and show the results in Table 3. The baseline model here is a model that only uses domain alignment loss without augmentation network. In fact, compared with this baseline, using all different variants of the augmentation network achieves performance improvement in spite of the artifacts in the generated images. But a more stable improvement is observed for the proposed augmentation network across different evaluation protocols. We also show more examples of augmented images in Figure 2.

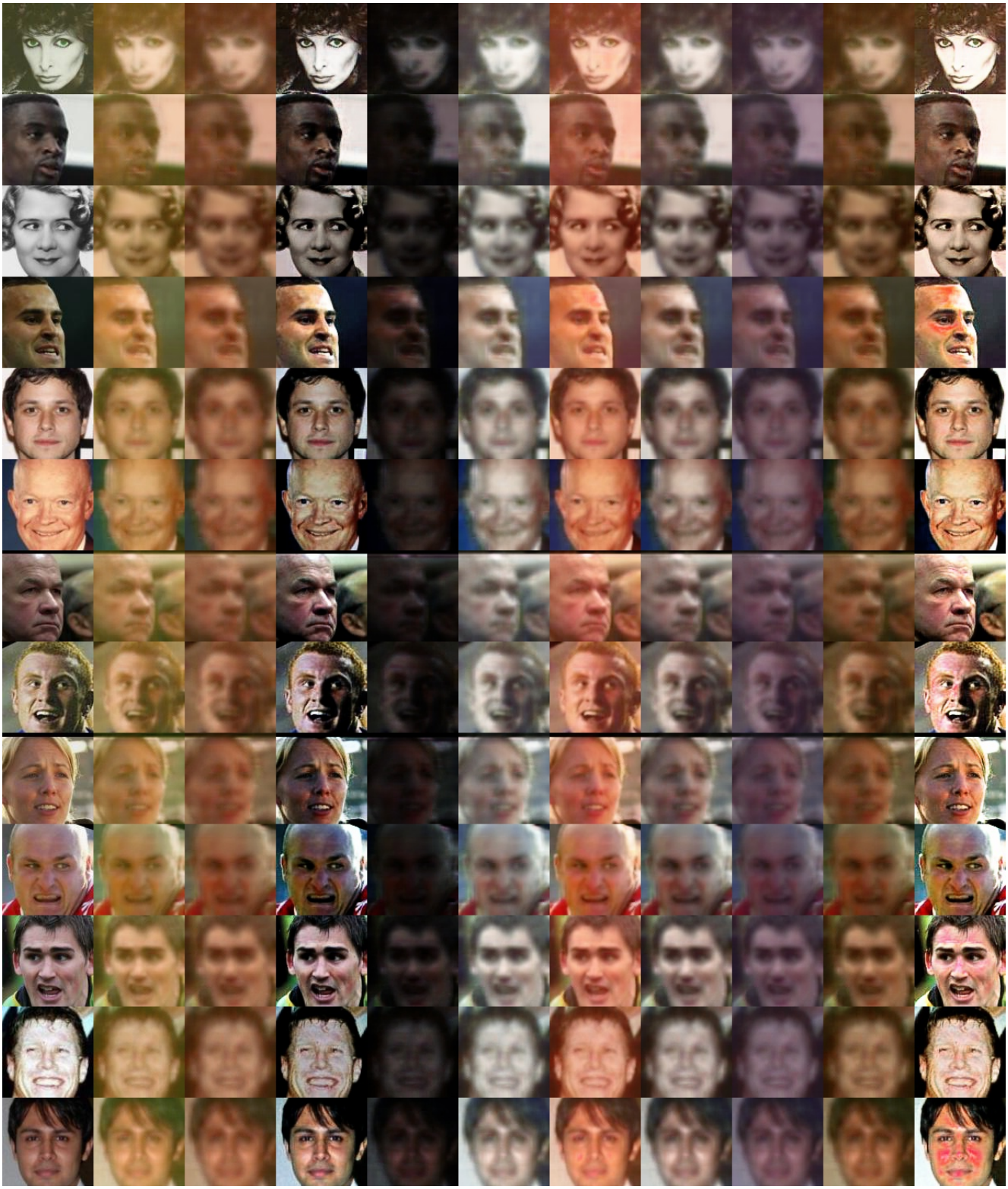

Figure 2: More examples of augmented images. The photos in the first column are the input images. The remaining images in each row are generated by the augmentation network with different style code.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv:1607.06450*, 2016. 1

[2] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *IEEE FG*, 2018. 1

[3] Jiankang Deng, Jia Guo, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. *CVPR*, 2019. 1

[4] Xun Huang and Serge J Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 1

[5] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 1, 2

[6] Bong-Nam Kang, Yonghyun Kim, Bongjin Jun, and Daijin Kim. Attentional feature-pair relation networks for accurate face recognition. In *ICCV*, 2019. 1

[7] Rajeev Ranjan, Ankan Bansal, Jingxiao Zheng, Hongyu Xu, Joshua Gleason, Boyu Lu, Anirudh Nanduri, Jun-Cheng Chen, Carlos D Castillo, and Rama Chellappa. A fast and accurate system for face detection, identification, and verification. *IEEE Trans. on Biometrics, Behavior, and Identity Science*, 2019. 1

[8] Yichun Shi and Anil K Jain. Probabilistic face embeddings. In *ICCV*, 2019. 1

[9] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv:1607.08022*, 2016. 1

[10] Weidi Xie, Li Shen, and Andrew Zisserman. Comparator networks. In *ECCv*, 2018. 1