HIME: Efficient Headshot Image Super-Resolution with Multiple Exemplars Supplementary Material

Xiaoyu Xiang[1,2*], Jon Morton[2], Fitsum A. Reda[2†], Lucas D. Young[2†], Federico Perazzi[2†],
Rakesh Ranjan[2], Amit Kumar[2], Andrea Colaco[2†], Jan P. Allebach[1]
[1]Purdue University, [2]Meta Reality Labs
{xiang43,allebach}@purdue.edu, {jamorton,rakeshr,akumar14}@meta.com,
{fitsum.reda,lucasyoung482}@gmail.com, fdp@bendingspoons.com, andrea@andreacolaco.info

Contents

A. Experimental Details

In this section, we first illustrate the architectures of our framework and the discriminator in Section A.1. Then, we present the objective functions for training them in Section A.2. Additional implementation details are described in Section A.3.

A.1. Architecture

HIME (small) The overall structure of this network is described in the main paper with the following components: LR feature extraction, HR feature extraction, RFA, CoFA, and HR reconstruction module. We included a large model and a small model in this paper, which are different in terms of the RFA module design.

For the large model, we used the flow-guided deformable alignment for RFA, which is described in detail in the main paper. For the small model, instead of using optical-flow to pre-align the features and guide the offset estimation, we directly use the LR and ref features to estimate the offsets, and align the reference features using deformable sampling:

$$F_i^{refA} = T(F_i^{ref}, \Phi_i) = DConv(F_i^{ref}, \Delta p_i). \tag{1}$$

where F_i^{refA} denotes the i-th aligned reference feature, $T(\cdot)$ is the sampling function, and Φ^i is the corresponding sampling parameters. The offset for the deformable sampling function are learned based on the similarity between the reference and input LR features. To make these features comparable, we take F_i^{refL} and F^L, which are both in the LR space, to predict the offset Δp_i for sampling the F_i^{ref}:

$$\Delta p_i = g([F_i^{refL}, F^L]), \tag{2}$$

where Δp_i also refers to the sampling parameter Φ_i; $g(\cdot)$ denotes a general operation of convolution layers for the offset estimation; $[\cdot, \cdot]$ denotes channel-wise concatenation. Please see Figure 1 for the detailed structure.

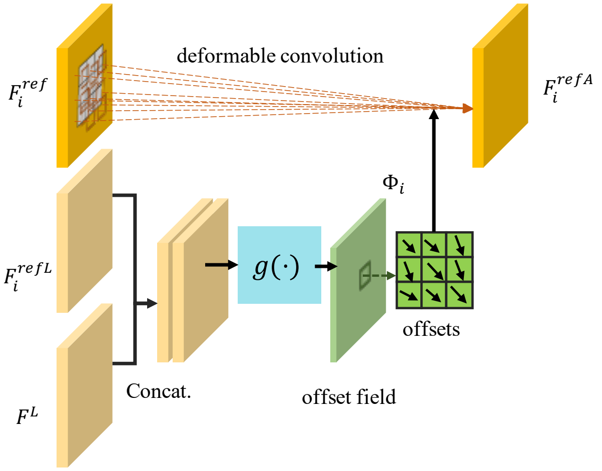

Figure 1: The architecture of RFA used in small models. It is purely based on deformable sampling.

Discriminators We use the discriminator in StyleGANv2 [4] as the architecture for the discriminator in our

[*] This work is done during the author's internship at Meta.
[†] Affiliated with Meta at the time of this work.

framework. It includes a convolutional layer and several residual blocks that downsample the input feature into different scales and turn it into an output tensor as the final prediction.

A.2. Objective Functions

The loss for training the perception-oriented models is composed of four parts: the pixel-wise reconstruction loss L_{rec}, the adversarial loss L_{adv}, the perceptual loss L_{per}, and our proposed correlation loss L_{cor}:

$$\mathcal{L}_P = \lambda_{rec} L_{rec} + \lambda_{adv} L_{adv} + \lambda_{per} L_{per} + \lambda_{cor} L_{cor}, \quad (3)$$

where λs are the weights for each loss term. In our implementation, $\lambda_{rec} = 1.0, \lambda_{adv} = 0.1, \lambda_{per} = 0.01, \lambda_{cor} = 0.1$. The pixel-wise reconstruction loss and the correlation loss are already described in the main paper. For the perceptual loss, we adopt the structure of VGG-19 [9] and extract the features Fea before the ReLU layer. The perceptual loss is measured by L_1 distance:

$$L_{per} = ||Fea^{HR} - Fea^{SR}||_1. \quad (4)$$

We adopt the relativistic GAN [3] for the L_{adv}:

$$L_{adv} = -\mathbb{E}_{HR}[\log(1 - D_{Ra}(I^{HR}, I^{SR}))] - \mathbb{E}_{SR}[\log(D_{Ra}(I^{SR}, I^{HR}))], \quad (5)$$

where I^{HR} and I^{SR} stand for the ground-truth and generated images, respectively. D_{Ra} denotes the relativistic average discriminator, which can be formulated as:

$$D_{Ra}(I^{HR}, I^{SR}) = \sigma(C(I^{HR}) - \mathbb{E}_{SR}[C(I^{SR})]), \quad (6)$$

where $C(\cdot)$ is the discriminator output, and σ is the Sigmoid function, $\mathbb{E}_{SR}[\cdot]$ stands for averaging all I^{SR} in a minibatch. The discriminator loss is defined as:

$$L_D = -\mathbb{E}_{HR}[\log(D_{Ra}(I^{HR}, I^{SR}))] - \mathbb{E}_{SR}[\log(1 - D_{Ra}(I^{SR}, I^{HR}))]. \quad (7)$$

A.3. Implementation Details

We generate LR inputs by bicubic downsampling with factor$= s$. For each LR input, we randomly select three different HR images to build the reference set during training. In the evaluation stage, we randomly select reference images to form an evaluation list. This list is applied to all evaluated methods for a fair comparison. The corresponding HR image of $s\times$ size is used for supervision. We use the Adam [5] optimizer, decaying the learning rate with a cosine annealing schedule for each batch [7] starting from 1×10^{-4}. For 16×16 LR inputs, we set the batch size as 128 and train the network on 1 Nvidia P100 GPU for 8×10^4 iterations. Our network is implemented with PyTorch [8].

B. Experimental Results

We further evaluate the influence of the difference between the reference and input images by choosing different poses in Sec. B.1. Further experiments with different thresholds of exemplar numbers are discussed in Sec. B.2. We discuss the choice of channels in the Ref branch in Sec. B.3 and evaluate the identity preservation of different methods in Sec. B.4. We also discuss the influence of face chirality under different data augmentation strategies in Sec. B.5.

B.1. Influence of Reference Images

In this section, we further evaluated the influence of the *pose* difference between the reference and the input. To examine the pose similarity between images, we choose the Multi-PIE dataset [2], which contains more than 750,000 images of 337 people under 15 view points while displaying a range of facial expressions. We downsample the Multi-PIE test data to construct 16×16 and 128×128 pairs, and test our $8\times$ models. We select three different views as reference images: frontal view (same pose), profile pose 1 and profile pose 2 (different pose). The pose similarity among all the three poses are: frontal $>$ profile 1 $>$ profile 2. Unlike frontal views, some facial components are missing in the profile faces. We evaluate our model with a single reference image for simplicity. From the results in Table 1, we can summarize that using the same pose achieves the best performance, benefiting from the high similarity between inputs and references. As for profile faces, the more similar view contributes better results, which is consistent with our observation.

Reference Pose	PSNR	SSIM
Same pose	24.360	0.7304
Profile pose - 1	24.352	0.7304
Profile pose - 2	24.346	0.7303

Table 1: Influence of pose difference between input and reference images.

B.2. Changing the Threshold of Exemplar Numbers

In the main paper, we construct the multi-exemplar training and testing data by removing the identities with < 4 exemplars. Still, our proposed method can handle an arbitrary number of exemplars. To verify if removing the few-shot samples introduces bias of the dataset, we change the threshold of exemplar number to 1, which adds 1629 identities with 3935 images. Based on the new data, we compared the performance on the new test data to verify if the proposed method benefit from the previous high-threshold, as shown in Tab. 2. We can observe that our method still benefit from the fewer number of exemplars compared to the non-ref baseline. Besides, we also train the network with the new training data and show the results in the last row of Tab. 2. This larger training set improves the PSNR by 0.21 and SSIM by 0.0092.

(LR, s)	Methods	PSNR	SSIM
(16, 8)	Bicubic	21.87	0.5912
	non-ref	23.84	0.7087
	HIME$_{rec}$(small)	24.55	0.7410
	HIME$_{rec}$ thr=1(small)	**24.76**	**0.7502**

Table 2: Results on the new test set with exemplar threshold=1.)

B.3. Color of the Reference Images

Intuitively, we expect the output image should keep consistent color with the LR input, instead of the color from the reference images. To justify the importance of exemplars' colors, we conducted an ablation study: comparing with directly extracting the features from Refs (24.35/0.732), converting the reference images to mono-channel before feature extraction shows almost no difference (24.34/0.732). While on the other hand, saving the high-resolution images in mono-channel can save the storage space as well as reduce the number of parameters and computational cost of the network. Thus, in our proposed network, the reference images are first converted from the RGB color space into a mono-channel feature map for a better balance between the performance and efficiency.

B.4. Evaluate the preservation of identity.

To evaluate if the reconstructed images can preserve the identity information, we adopt the SOTA face recognition model ArcFace [1] to calculate the cosine similarity of the identity features between the ground-truth images and the ones generated by HIME (small), and show the results in Tab.3. A higher similarity indicates the identity info is closer to GT. From this table, we can observe that our method performs better than the other SOTA methods in maintaining the identity information, especially HIME$_{rec}$. We thank for the constructive feedbacks and include the new experiments in Appendix B.5.

B.5. Influence of Face Chirality

Typical human faces contain a variety of asymmetries. [6] brings up the visual chirality in faces and the distribution bias in public face datasets. Here we compare the influence of such asymmetries in headshot RefSR to answer the following questions: 1) Does the mismatch between input LR and references matter? 2) Does the bias in the training set influence the reconstruction performance?

Methods	Bicubic	PFSR	FSRNet	GWAINet
$s_{id}\uparrow$	0.1126	0.3029	0.2082	0.1321
Methods	SPARNet	PSFR-GAN	HIME$_{rec}$	HIME$_p$
$s_{id}\uparrow$	0.1226	0.2692	**0.4408**	0.4045

Table 3: Comparison of identity similarity.

Models	No-aug	Uneven h-flip	Even h-flip	PSNR↑	SSIM↑
(a)	√			22.60	0.660
(b)		√		22.52	0.654
(c)			√	23.63	0.662

Table 4: Influence of face chirality reflected by different augmentation strategies.

Tabel 4 shows our experimental results of changing the augmentation: (a) no augmentation; (b) randomly horizontal-flip the LR or Ref images, but not both for a given pair, which introduces the face view mismatch; (c) randomly horizontal-flip both the LR and reference images, which balances the number of left and right faces without introducing mismatches. By comparing (a) and (b), we can observe that the PNSR drops by 0.08 and the SSIM drops by 0.006, respectively, which indicates that training with mismatched views of faces would impair the model's performance. Compared with (a), (c) performs better in terms of the PSNR and SSIM by a small margin, which demonstrates that the proper augmentation improves the performance by mitigating the distribution bias in the dataset.

References

[1] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 3

[2] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. In *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–8, 2008. 2

[3] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard gan. *arXiv preprint arXiv:1807.00734*, 2018. 2

[4] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 1

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[6] Zhiqiu Lin, Jin Sun, Abe Davis, and Noah Snavely. Visual chirality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12295–12303, 2020. 3

[7] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 2

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32:8026–8037, 2019. 2

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2