Identifying Missing Children: Face Age-Progression via Deep Feature Aging (Supplementary Material)

In this supplementary material, we conduct thorough experiments to show (a) why deep feature aging works, (b) the effects of deep feature aging on face embeddings, and (c) ablation on style-dimensionality. We include more implementation details on our Feature Aging Module (FAM) and the Image Generator used for aging face images. We also show additional examples of aged faces.

I. What is Feature Aging Module Learning?

A. Age-Sensitive Features

In order to analyze which components of the face embeddings are altered during the aging process, we first consider a decoder that takes an input a face embedding and attempts to construct a face image without any supervision from the image itself. That is, the decoder is a variant of the proposed Image Generator without the style encoder. This ensures that the decoder can only synthesize a face image from whatever is encoded in the face feature only.

We then compute a difference image between a reconstructed face image and its age-progressed version. We directly age the input feature vector via our Feature Aging Module as shown in Figure 1. We find that when a probe feature is progressed to a younger age, our method attempts to reduce the size of the head and the eyes, whereas, age-progression enlarges the head, adds makeup, and adds aging effects such as wrinkles around the cheeks. As we expect, only components responsible for aging are altered, whereas, noise factors such as background, pose, quality, and style remain consistent.

B. Why does Deep Feature Aging enhance longitudinal performance?

In Figure 2, in the first row, we see that originally a state-of-the-matcher, CosFace [1], wrongly retrieves the true mate at Rank-37. Interestingly, we find that the top 5 retrievals are very similar ages to the probe's age (17 years). That is, state-of-the-art matchers are biased towards retrieving images from the gallery that are of similar ages as that of the probe. With our Feature Aging Module (FAM), we age the feature in the feature space of the matcher such that we can 'fool' the matcher into thinking that the gallery is closer to the probe's age. In this manner, the matcher tends to utilize identity-salient features that are age-invariant and can only focus on those facial components. In row 2, we find that when we age the gallery to the probe's age, the top retrievals are all children. This highlights the strength of our Feature Aging Module and its ability to enhance longitudinal performance of matchers.

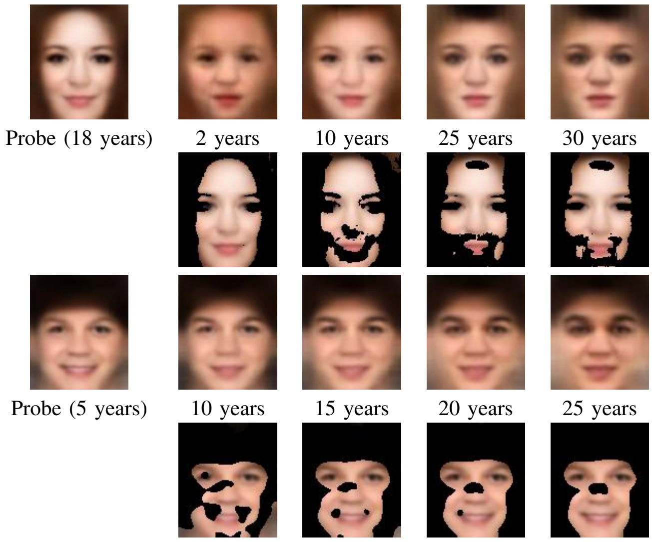

Fig. 1: Differences between age-progressed and age-regressed features. Rows 1 and 3 show decoded aged images for two different individuals. Rows 2 and 4 show the areas that change from the probe (black color indicates no-changes). For both progression and regression, our proposed Feature Aging Module only alters face components responsible for aging while largely keeping other covariates such as background, pose, quality, and style consistent.

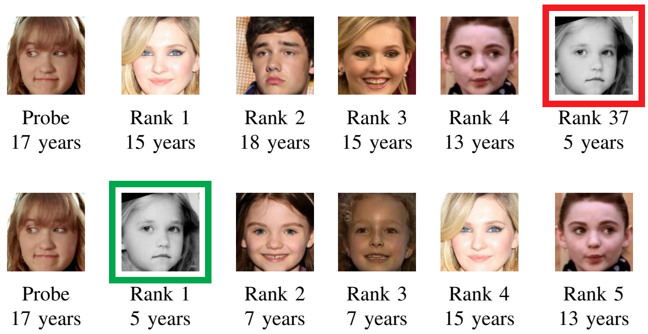

Fig. 2: Row 1: CosFace wrongly retrieves the true mate at Rank-37. Top-5 retrievals include gallery images that are of similar ages as that of the probe. Row 2: With the proposed Feature Aging Module, CosFace focuses only on identity-salient features that are age-invariant and retrieves children in the top-5 retrievals. In this scenario, our Feature Aging Module can aid CosFace in retrieving the true mate at Rank-1.

C. Effect of Deep Feature Aging on Embeddings

To observe the effect of our module on the face embeddings, we plot the difference between the mean feature vectors of all subjects (in the test set) at the youngest age in the CFA dataset, *i.e.* 2 years, and mean feature vectors at different target ages (in the test set) (see Figure 3).

For a state-of-the-art face matcher, CosFace [1], the differences between these mean feature vectors, over all 512 dimensions, increases over time lapse causing the recognition accuracy of the matcher to drop for larger time lapses. However, with the proposed feature aging module, the difference remains relatively constant as the time lapse increases. This indicates that the proposed feature aging module is able to maintain relatively similar performance over time lapses.

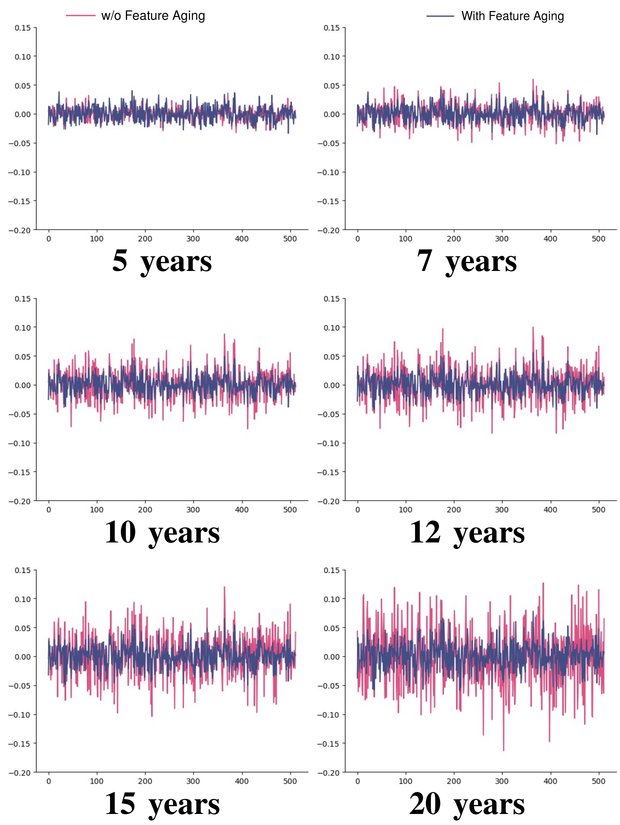

Fig. 3: Difference between mean feature at age 2 years and different target ages (denoted below each plot) with CosFace [1] and CosFace [1] with the proposed feature aging module on the CFA dataset. The difference between the mean features increases as the time lapse becomes larger but with the proposed feature aging module the difference is relatively constant over time lapse. This clearly shows the superiority of our module over the original matcher.

II. Effect of Style Dimensionality

In this section, we evaluate the effect of increasing or decreasing the dimensionality of the style vector obtained via the proposed Style Encoder. Given a style vector of k-dimensions, we concatenate the style vector to a d-dimensional ID feature vector extracted via an ID encoder. For this experiment, we consider a 512-dimensional ID embedding obtained via CosFace [1].

We evaluate the identification rate when the gallery is aged to the probe's age as well simply reconstructing the gallery to its own age. We train our proposed framework on ITWCC training dataset [2] and evaluate the identification rate on a validation set of CLF dataset. Note that we never conduct any experiment on the testing sets of either ITWCC nor CLF datasets. In Figure 4, we observe a clear trade-off between reconstruction accuracy and aging accuracy for $k = 0,\ 32,\ 128,\ 512,\ 1024$. That is, for larger values of k, the decoder tends to utilize more style-specific information while ignoring the ID feature (which is responsible for aging via FAM). In addition, the gap between reconstruction and aging accuracy narrows as k gets larger due to the decoder ignoring the identity feature vector from the ID encoder. In Figure 5, we can observe this trade-off clearly. Larger k enforces better perceptual quality among the synthesized images, with lesser aging effects and lower accuracy. Therefore, we compromise between the visual quality of the synthesized and accuracy of aged images and decided to use $k = 32$ for all our experiments. Note that, $k = 0$, can achieve nearly the same accuracy as the feature aging module alone, however, the visual quality is compromised. Therefore, an appropriate k can be chosen, depending on the security concerns and application.

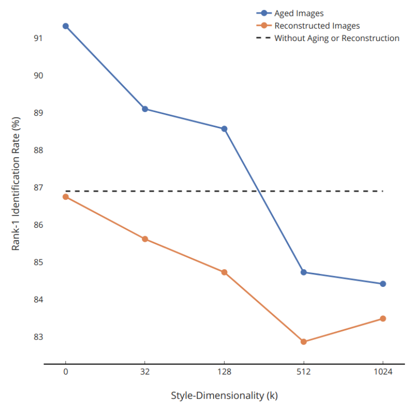

Fig. 4: Trade-off between identification rates from reconstruction and aging. For our experiments, we choose $k = 32$.

III. Implementation Details

All models are implemented using Tensorflow r1.12.0. A single NVIDIA GeForce RTX 2080 Ti GPU is used for training and testing.

A. Data Preprocessing

All face images are passed through MTCNN face detector [3] to detect five landmarks (two eyes, nose, and two mouth corners). Via similarity transformation, the face images are aligned. After transformation, the images are resized to 160×160 and 112×96 for FaceNet [4] and CosFace [1], respectively.

a) Feature Aging Module: For all the experiments, we stack two fully connected layers and set the output of each layer to be of the same d dimensionality as the ID encoder's feature vector.

b) Image Generator: All face images are cropped and aligned via MTCNN [3] and resized to 160×160. The style-encoder is composed of four convolutional layers and a fully connected layer in the last stage that outputs a k-dimensional style feature vector. In all our experiments, $k = 32$. The

Fig. 5: Trade-off between visual quality and aging. For our experiments, we choose $k = 32$.

decoder first concatenates the k-dimensional style vector and the d-dimensional ID vector from the ID encoder into a $(k+d)$-dimensional vector followed by four-strided convolutional layers that spatially upsample the features. All convolutional and strided convolutional layers are followed by instance normalization with a leaky ReLU activation function. At the end, the decoder outputs a $160 \times 160 \times 3$ image (for FaceNet) and $112 \times 96 \times 3$ image (for CosFace). We emperically set $\lambda_{ID} = 1.0$, $\lambda_{pix} = 10.0$, and $\lambda_{tv} = 1e - 4$.

We train the proposed framework for 200,000 iterations with a batch size of 64 and a learning rate of 0.0002 using Adam optimizer with parameters $\beta_1 = 0.5, \beta_2 = 0.99$. In all our experiments, $k = 32$.

IV. Visualizing Feature Aging

In Figures 6 and 7, we plot additional aged images via our proposed aging scheme to show the effectiveness of our feature aging module and image generator.

References

[1] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*, 2018. 1, 2, 4, 5

[2] Nisha Srinivas, Karl Ricanek, Dana Michalski, David S. Bolme, and Michael A. King. Face Recognition Algorithm Bias: Performance Differences on Images of Children and Adults. In *CVPR Workshops*, 2019. 2

[3] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE SPL*, 23(10):1499–1503, 2016. 2

[4] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, pages 815–823, 2015. 2

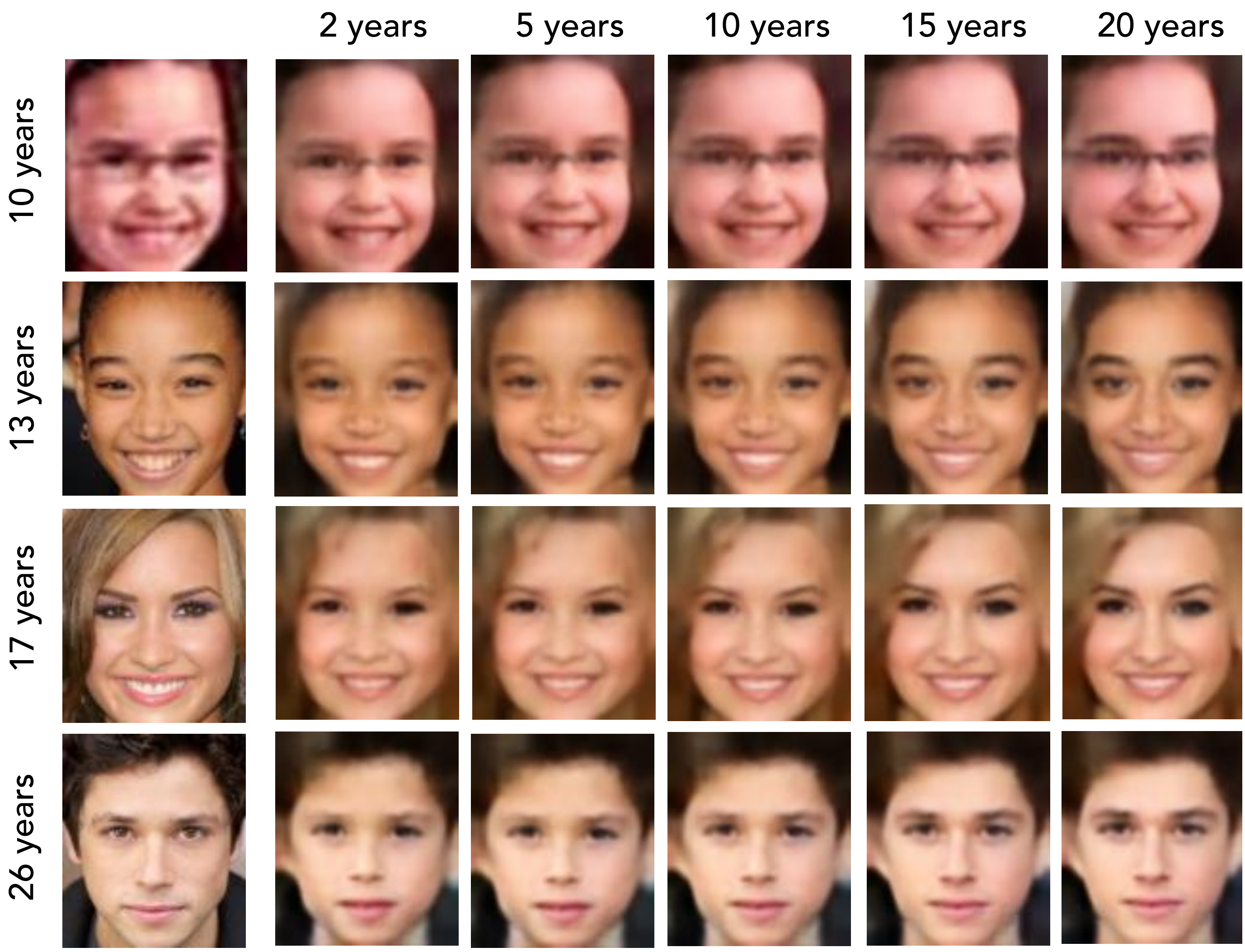

Fig. 6: Aged face images from the proposed aging scheme using CosFace [1] to specified target ages on ITWCC dataset.

Fig. 7: Aged face images from the proposed aging scheme using CosFace [1] to specified target ages on CFA dataset.